

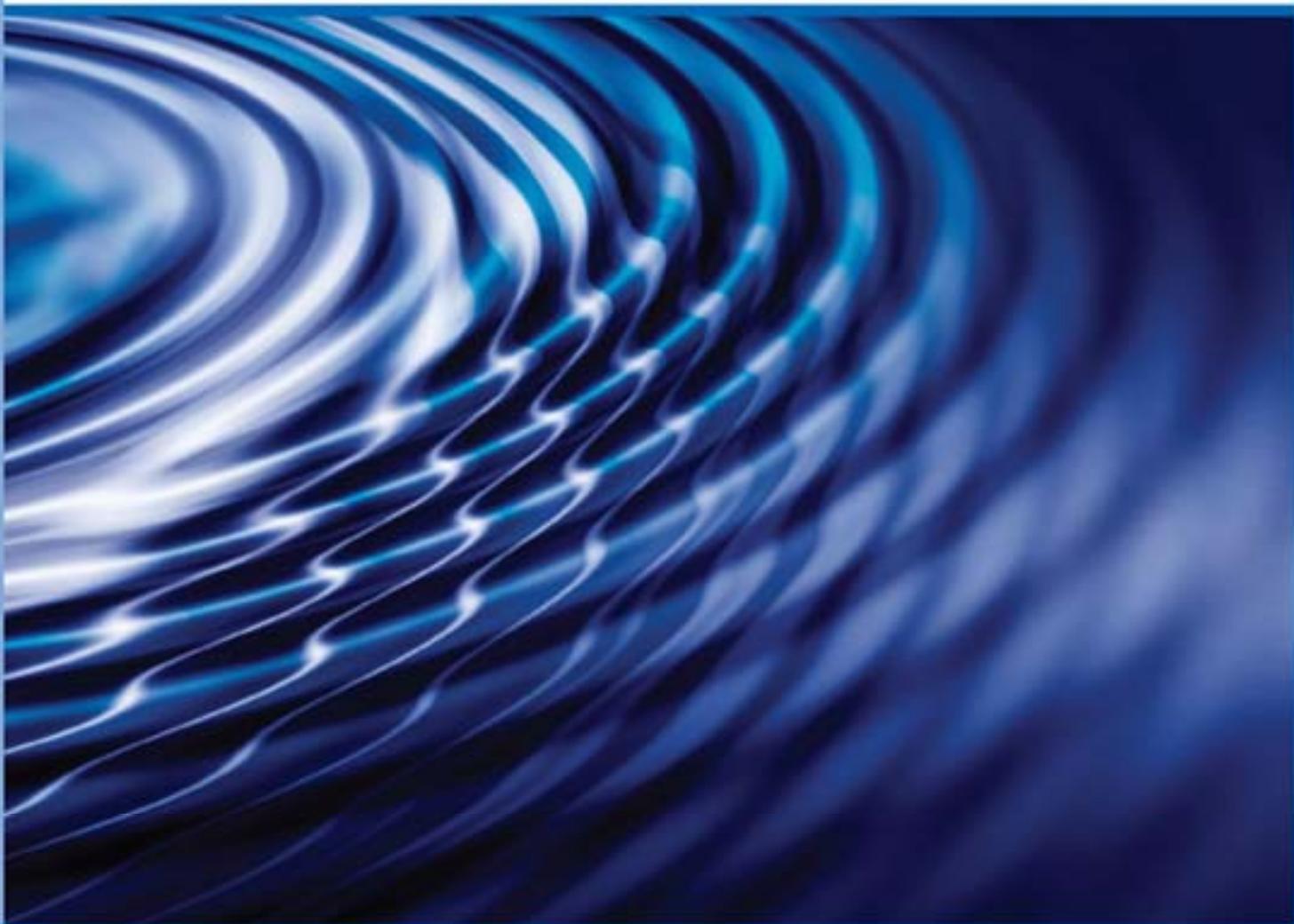
WINDOWS SERVER USER'S GUIDE



Attachmate®

Reflection®

for Secure IT



Reflection for Secure IT

Windows Server

Version 7.2



© 2010 Attachmate Corporation. All rights reserved.

No part of the documentation materials accompanying this Attachmate software product may be reproduced, transmitted, transcribed, or translated into any language, in any form by any means, without the written permission of Attachmate Corporation. The content of this document is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Attachmate Corporation. Attachmate Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this document.

Attachmate, the Attachmate logo, and Reflection are registered trademarks of Attachmate Corporation, in the USA. All other trademarks, trade names, or company names referenced herein are used for identification only and are the property of their respective owners.

Attachmate Corporation
1500 Dexter Avenue North
Seattle, WA 98109
USA
+1.206.217.7100
<http://www.attachmate.com>

Contents

Introduction	9
Installing Reflection for Secure IT	11
Install and Uninstall Reflection for Secure IT	11
Install and Uninstall Reflection PKI Services Manager	12
Upgrading from Reflection for Secure IT version 7.x	13
Automatic Migration of Reflection 6.x and F-Secure Settings	14
Uninstall Reflection for Secure IT	15
Installer Advanced Tab	16
Getting Started	17
Get Started with the Server Console	17
Start and Stop the Server	18
Getting Help	19
Understanding Secure Shell	19
General Server Configuration	21
Saving Server Settings	21
Restore Default Settings	22
Status Tab	22
General Pane	23
Change the Server Port	24
Network Pane	24
Network Binding Dialog Box	25
The Reflection for Secure IT Data Folder	26
Change the Default Data Folder	26
Using a Server Cluster	29
Configure a Reflection for Secure IT Cluster	30
Set Data Folder Dialog Box	33

Data Protection	35
Data Encryption	35
Data Integrity	36
Federal Information Processing Standard (FIPS)	36
Encryption Pane	37
Key Exchange	38
Key Exchange Pane	38
Server Authentication	41
Public Key Authentication	42
Configure Public Key Host Authentication	43
Identity Tab	44
Generate Host Private Key Dialog Box	46
Export Public Key Dialog Box	47
Certificate Authentication Overview	48
Configure Certificate Server Authentication	49
Windows Certificate Store Dialog Box	51
Kerberos (GSSAPI) Authentication	51
Configure GSSAPI Server and Client Authentication	52
User Authentication	55
Authentication Pane	56
Password and Keyboard Interactive Authentication	58
Configure Password User Authentication	58
Configure Keyboard Interactive User Authentication	59
Password Pane	60
Public Key Authentication for Users	61
Configure Public Key User Authentication: Reflection for Secure IT Windows Clients	61
Configure Public Key User Authentication: Reflection for Secure IT UNIX Clients	62
Public Key Pane	63

Certificate Authentication for Users	65
Using PKI Services Manager	66
Configure Certificate Authentication for Users	67
Certificates Pane	70
Start and Stop the PKI Services Manager Service on Windows	71
RSA SecurID Authentication	72
Configure SecurID Authentication	73
RSA SecurID Pane	74
RADIUS Authentication	74
Configure RADIUS Authentication	75
RADIUS Pane	76
RADIUS Server Dialog Box	77
GSSAPI (Kerberos) Authentication	77
Configure Client Authentication using Windows Credentials	77
GSSAPI / Kerberos V5 Pane	79
Cached Credentials	81
Understanding How Credentials Affect User Access to Resources	82
Best Practices for Using Cached Credentials	83
Record and Use Cached Credentials	86
Credential Cache Pane	87
Domain Access Pane	90
Add/Edit Credential Dialog Box	91
Select Credential Dialog Box	92
Filters Dialog Box	94
Mapped Drives	95
Configure Mapped Drives for Terminal Sessions	95
Mapped Drives Pane	97
Mapped Drive Settings Dialog Box	98

Secure File Transfer	99
File Transfer Overview	99
Specify the User Login Directory	100
Customize Directory Access for File Transfers	101
Control Upload and Download Access	103
Virtual Root Directory in Reflection for Secure IT	103
Smart Copy and Checkpoint Resume	106
SFTP Directories Pane	107
Accessible Directory Settings Dialog Box	109
Controlling Access	111
Access Control Settings	111
Using Allow and Deny Rules for Access Control	112
Controlling Access from Client Computers	114
Client Host Access Control Pane	115
Client Host Access Control Dialog Box	116
Controlling Access by Group	118
Group Access Control Pane	120
Group Access Control Dialog Box	121
Controlling Access by User	122
User Access Control Pane	123
User Access Control Dialog Box	124
Command Shell Access	125
Permissions Pane	125

Working with Subconfigurations	129
Subconfiguration Overview	129
Configure Settings Specific to a Client Host	130
Client Host Configuration Pane	131
Client Host Configuration Dialog Box	131
Configure Group-Specific Settings	133
Group Configuration Pane	134
Group Configuration Dialog Box	134
Configure User-Specific Settings	135
User Configuration Pane	136
User Configuration Dialog Box	136
Revert settings to inherited values	138
Port Forwarding	139
Port Forwarding Overview	139
Disable Port Forwarding	140
Troubleshooting	141
Logging	141
Use the Windows Event Viewer	141
Event Logging Pane	142
Enable Logging to a Text File	143
Debug Logging Pane	143
Custom Log Events Dialog Box	145
Managing System Resources	146
Troubleshooting Group Settings	146
Troubleshooting Reflection for Secure IT Help	147

Appendix	149
Files Used by Reflection for Secure IT	150
Regular Expression Syntax	152
Table of Migrated Settings	154
Table of Migrated PKI Settings	162
Manual Host Key Migration	164
Pattern Strings in Directory Paths	165
Keyboard Access to Console Features	167
winpki and pkid Command Reference	168
pkid_config Configuration File Reference	172
pki_mapfile Map File Reference	178
Sample Mapping Rules	185
Sample Map File with RuleType Stanzas	187
PKI Services Manager Return Codes	188
rsshd Command Line Utility	191
ssh-keygen Command Line Utility	193
ssh-certtool Command Reference	196
Glossary of Terms	201
Index	209

Introduction

Reflection for Secure IT Windows Server is a full-featured, easily customizable Windows-based Secure Shell server.

With Reflection for Secure IT, you can:

- Support secure file transfer using the Secure Shell protocol
- Control and customize file transfer directories
- Monitor the number of connected sessions
- Configure all server settings using the server console
- Configure public key, certificate, or GSSAPI host authentication
- Configure password, keyboard interactive, public key, certificate, RADIUS, SecurID, and GSSAPI/Kerberos user authentication
- Specify which encryption, hashing, and key exchange algorithms the server supports
- Enforce FIPS140-2 algorithm standards
- Control access to the server from client hosts, groups, or users
- Configure customized settings for client hosts, individual users, or user groups
- Audit and troubleshoot using configurable logging information
- Use command-line utilities to control the server and manage keys and certificates
- Configure the server to run in a clustered environment

CHAPTER 1

Installing Reflection for Secure IT

In this Chapter

Install and Uninstall Reflection for Secure IT	11
Install and Uninstall Reflection PKI Services Manager	12
Upgrading from Reflection for Secure IT version 7.x	13
Automatic Migration of Reflection 6.x and F-Secure Settings	13
Uninstall Reflection for Secure IT	15
Installer Advanced Tab	16

Install and Uninstall Reflection for Secure IT

For information about supported platforms, and additional system requirements, see *Technical note 1944* (<http://support.attachmate.com/techdocs/1944.html>).

Note: You must be logged in as an Administrator to install, configure, and run Reflection for Secure IT Windows Server.

To install Reflection for Secure IT Windows Server from the download site

- 1 Click the download link and run the download program.
- 2 Select a location for the installer files and click **OK**.

The files are extracted to the specified location, and the Attachmate Setup wizard starts.
- 3 (Optional) To personalize the installation, click the **User Information** tab and enter the name, organization, and Volume Purchase Agreement (VPA) number (if you have a VPA).

Note: VPA numbers are used by customer support to expedite service requests.

- 4 (Optional) To change the default installation folder, click the **File Location** tab and browse to the folder in which you want to install Reflection for Secure IT.

12 Reflection for Secure IT

- 5 Click **Install Now**.
- 6 The server is not fully functional until you restart your computer. To reboot immediately, select **Restart my computer for me** and click **Close**.

Note: After a restart, the server runs automatically. A Windows Service called "Attachmate Reflection for Secure IT Server" is created with an automatic startup type.

To uninstall Reflection for Secure IT

- 1 Log in as an administrator.
- 2 From the Windows **Programs and Features** (or the **Add or Remove Programs**) control panel, select Reflection for Secure IT Windows Server Server.
- 3 Click **Uninstall** (or **Remove**).

Install and Uninstall Reflection PKI Services Manager

Reflection PKI Services Manager is a service that provides X.509 certificate validation services. If you need support for user certificate authentication, you'll need to download and install this application. It is available at no additional charge.

To install Reflection PKI Services Manager

- 1 Log in as an administrator.
- 2 Download the Reflection PKI Services Manager installer package.
- 3 Run the **Setup** program.

Note: To install the service, accept the default settings on the **Advanced** tab. Creating an administrative installation image does not actually install the product — instead, it places the install files on a network location for later installation to multiple workstations.

- 4 *Start the service* (page [71](#)).

Notes:

- On Windows, starting the console or the service for the first time initializes PKI Services Manager. This creates the required data folders and default settings files. If these folders already exist, they are not changed; PKI Services Manager uses your existing data files and folders. (On UNIX the install script automatically initializes PKI Services Manager if required, and starts the service.)
 - Before Reflection PKI Services Manager can validate certificates you need to edit the default configuration and map files.
-

To uninstall Reflection PKI Services Manager

- 1 Log in as an administrator.
- 2 From the Windows **Programs and Features** (or the Add or Remove Programs) control panel, select Attachmate Reflection PKI Services Manager.
- 3 Click **Uninstall** (or **Remove**).

Upgrading from Reflection for Secure IT version 7.x

You can install this version over an existing earlier version of Reflection for Secure IT version 7.x. The upgrade replaces the existing version. If you are upgrading an existing copy, note the following:

- We recommend that you back up your server configuration file before upgrading. This may be useful if you want to revert to an earlier version at some point in the future.
- After applying the upgrade, you need to restart Windows to complete the installation.

Automatic Migration of Reflection 6.x and F-Secure Settings

When you install Reflection for Secure IT Windows Server on systems with an F-Secure server or Reflection for Secure IT version 6.x, Reflection for Secure IT automatically migrates your current identity (host key and certificates) and settings. Your existing key and configuration files are not changed.

- Existing host keys (`hostkey` and `hostkey.pub` by default) are copied to the new *key location* (page 205), so you don't need to make any changes to clients that are configured to trust your current host key.
- Settings in your existing `sshd2_config` file are migrated to the new *xml configuration file* (page 204). Migration information is saved to the *migration log file* (page 205).
- If you used a password cache, cached passwords are migrated to the new password cache file.

This migration occurs the first time you:

- Start the server console. This triggers the migration of keys and settings without automatically starting the server.

-or-

- Start the service. When you restart Windows, the service starts automatically. This triggers the migration and starts the server using the migrated key and settings. (You can also start the service manually using the `rsshd` command line or using the Windows Computer Management console.) Note: The service cannot start if an earlier version server is still running using the same port.

Notes:

- It is possible to run both version 7.x and 6.x on the same computer. If you want to test version 7.x before uninstalling the earlier version, either stop the earlier version service, or configure version 7.x to use a different port.
 - If you have an existing XML settings file, the server will not migrate the settings from a previous version settings file. This enables you to configure a single settings file and install it onto multiple servers.
 - Automatic migration won't take place if you have already uninstalled your prior version.
 - You can manually migrate settings using the `rsshd` command line utility with the `-m` option.
-

Automatic Migration of PKI Settings

Settings for validating user certificates are configured in the `sshd2_config` file in F-Secure and Reflection for Secure IT version 6.x. Starting with version 7.1, user authentication with certificates is supported by Reflection PKI Services Manager. When you first start PKI Services Manager on a system that has a prior version `sshd2_config` file, certificate authentication settings are automatically migrated to the `pki_config` and `pki_mapfile` files used by PKI Services Manager.

Notes:

- If the `pki_config` file in the destination folder already has a trust anchor configured, no migration occurs. This helps ensure that the migration won't overwrite modifications you have already configured.
 - You can manually migrate PKI settings using the **winpki** command line utility with the **-m** option.
-

Uninstall Reflection for Secure IT

Notes:

- You must be logged in as an Administrator to uninstall the server.
 - To complete the server uninstall, you must restart Windows.
 - Uninstalling the server does not remove the host key or any other files created in the server *data folder* (page [205](#)).
-

To uninstall the server

- 1 From the **Windows Control Panel**, open **Add or Remove Programs** (or **Programs and Features**).
- 2 Select **Attachmate Reflection for Secure IT Server**, and then click **Remove**.
- 3 Restart Windows.

To remove the host key and additional files created by Reflection for Secure IT

- Delete the server *data folder* (page [205](#)), and all of its contents.

Note: To permanently delete files containing sensitive data, such as the host key file and log files, you may want to use a secure deletion utility.

Installer Advanced Tab

Note: Creating an administrative installation image does not actually install the product — instead, it places the install files on a network location for later installation to multiple workstations.

The options are:

Install to this PC	Installs Reflection for Secure IT to your computer.
Create an Administrative Install image on a server	When you create an administrative installation image, an image of Reflection for Secure IT is copied to a network location for later installation to multiple workstations. This network location can be used by deployment tools to access and create packages that are deployed to workstations. Also, end users can perform installations by running <code>setup.exe</code> from this location.
Create a log file for this installation	Specifies whether or not a log file is created. The installation log file, which provides details about the installation, is saved in the user's temp directory (<code>%tmp%</code>) with a generated name that begins with <code>atm</code> . To open this folder, launch the Start menu Run command and enter <code>%tmp%</code> .

CHAPTER 2

Getting Started

In this Chapter

Get Started with the Server Console	17
Start and Stop the Server	18
Getting Help	19
Understanding Secure Shell	19

Get Started with the Server Console

Use the server console to configure the server, and start or stop the service. All Reflection for Secure IT settings can be configured using the server console.

To work with the server console

- 1 On the Windows Start menu, click **Attachmate Reflection > Reflection SSH Server Configuration**.

The server automatically creates (or migrates) a host public/private key pair, and is configured to use this key pair for host authentication.

- 2 Click **Identity** to view information about the default host private key. You can use this tab to change the host key, or configure host authentication using a digital certificate.
- 3 Click **Configuration** to view and edit the current server settings.
- 4 For more information about the settings on a server pane, click  on that pane.
- 5 Click **File > Save Settings** to save any changes to the server configuration file.
- 6 Click **Action > Start Server** to start the service.
- 7 Click **Status** to view the server status, uptime, and number of client connections.

Notes:

- You must be logged in as an Administrator to configure and run Reflection for Secure IT.
 - If you install the server and restart Windows before you run the server console, Reflection automatically creates (or migrates) a host key pair at that time, and starts the server using default settings.
-

Start and Stop the Server

Use the console to start and stop the server.

To start the server

- Restart Windows.

The server is installed as a service and starts automatically whenever you start Windows.

-or-

- Launch the server console and click **Action > Start Server**.

-or-

- Click the toolbar Start button:



To stop the server

- Launch the server console and click **Action > Stop Server**.

-or-

- Click the toolbar Stop button:



To restart the server

- On the **Action** menu, click **Restart Server**.

-or-

- Click the toolbar Restart button:



A restart stops and restarts the server. If you have changed the server configuration, restarting the server ensures that all client connections use the new settings. Without a restart, new client sessions use the new settings, but existing connections continue to use the settings that were in effect when those connections were established.

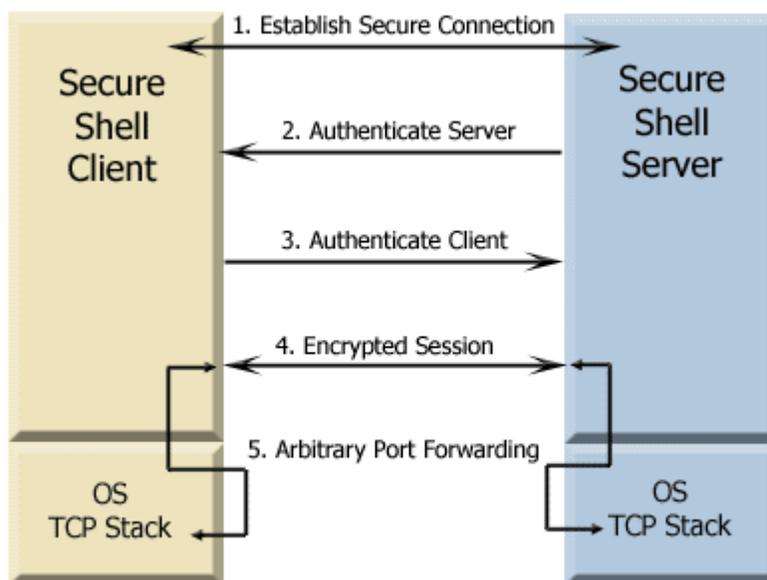
Getting Help

For help configuring and running Reflection for Secure IT, use the following:

Help menu	From the server console, go to Help > Help Topics to view the complete help system.
Context-sensitive help	Click the help icon (🔗) from any pane (or press F1) to view a topic with information specific to that pane.
Help online	Visit http://support.attachmate.com/manuals/ to view Attachmate help content and manuals online.
Technical notes	Visit http://support.attachmate.com/techdocs/ to view the Attachmate technical note library.

Understanding Secure Shell

This diagram outlines the basic steps involved in creating a Secure Shell channel and using it to transmit data securely.



1. **Establish a secure connection.**

The client and server negotiate to establish a shared key and cipher to use for session encryption, and a hash to use for data integrity checking. For additional information, see *Data Protection* (page [35](#)).

2. **Authenticate the server.**

Server authentication enables the client to confirm the identity of the server. The server has only one chance to authenticate to the client during the authentication process. If this authentication fails, the connection fails. For additional information, see *Server Authentication* (page [41](#)).

3. **Authenticate the client.**

Client authentication enables the server to confirm the identity of the client user. By default, the client is allowed multiple authentication attempts. The server and client negotiate to agree on one or more authentication methods. For additional information, see *Client Authentication* (page [55](#)).

4. **Send data through the encrypted session.**

Once the encrypted session is established, all data exchanged between the Secure Shell server and client is encrypted. Users now have secure remote access to the server and can execute commands and transfer files securely through the secure channel. For information about configuring secure file transfer, see *Secure File Transfer* (page [99](#)).

5. **Use port forwarding to secure communications between other clients and servers.**

Port forwarding, also known as tunneling, provides a way to redirect communications through the Secure Shell channel of an active session. When port forwarding is configured, all data sent to a specified port is redirected through the secure channel. For additional information, see *Port Forwarding* (page [139](#)).

CHAPTER 3

General Server Configuration

In this Chapter

Saving Server Settings	21
Restore Default Settings	21
Status Tab	22
General Pane	22
Change the Server Port	23
Network Pane	24
Network Binding Dialog Box	24
The Reflection for Secure IT Data Folder	25

Saving Server Settings

As soon as you start the server, it is ready to manage connections using default settings. If you want to change the default configuration, use the server console. Settings changes take effect when you update the settings file using **File > Save**. Existing connections remain active using their original settings; subsequent connections use the new settings.

Settings are saved in the *server configuration file* (page [204](#)). This file is in XML format.

Note: To minimize the possibility of introducing errors, we strongly recommend using the console whenever you want to modify your server settings, rather than editing the XML configuration file directly.

Restore Default Settings

You can reset defaults for all of the server settings, or for just the currently displayed configuration pane.

To reset all settings to default values

- 1 From the **Action** menu, click **Restore All Default Settings**.
- 2 Click **Yes** in response to the confirmation prompt.

Note: This procedure saves a numbered backup copy of your settings file (for example, `rsshd_config.xml.001`).

To reset defaults in the currently displayed pane

- 1 From the **Action** menu, click **Restore Pane Defaults**.

-or-

From the current pane, click **Restore pane defaults** ()

- 2 Click **Yes** in response to the confirmation prompt.
- 3 Save your settings (**File** > **Save Settings**).

Status Tab

Getting there

- The **Status** tab displays by default when you launch the server console.

The options are:

Status	Shows whether the server is running or stopped.
Uptime	Shows how long the server has been running.
Open Connections	Shows the number of client connections to the server.

General Pane

Getting there

- From the server console, click **Configuration** > **General**.

The options are:

Maximum number of connections

Sets the maximum number of client connections allowed. Sessions that reuse an existing tunnel are considered a single connection. Use zero (0) to set no limit. In this case, limits set by the operating system may affect the number of possible connections.

Note: You can use **Maximum number of connections** to avoid exhausting the Windows system memory available for creating desktop heaps. For more information, see *Managing System Resources* (page [146](#)).

Maximum connections per user

Sets the maximum number of connections allowed per user. Use zero (0) to set no limit.

Note: If a client opens multiple sessions using the client connection reuse feature, these sessions count as a single connection.

Session time-out (seconds)

Sets the number of seconds a connection can remain inactive before the server terminates the connection. If a connection tunnel is shared by multiple sessions, all sessions must remain idle for this duration. Use zero (0) to set no limit.

Banner message file

Identifies a file that contains text for a banner message. The server sends this text to the client before the client authenticates. Use of a banner is necessary in some jurisdictions to ensure legal protection against violators. If the file is in a format other than UTF-8, it is automatically converted to UTF-8.

Note: Some clients do not support banner display. If you configure a banner, you should ensure that your Secure Shell client supports this feature.

Change the Server Port

By default, the server is configured to listen for connections from all network adapters on port 22.

To change the server port

- 1 Start the server console, and then click **Configuration**.
- 2 Click **Network**.
- 3 Select the listening address you want to modify, and then click **Edit**.

Note: Select `::` to modify the IPv6 listening address and `0.0.0.0` to modify the IPv4 listening address.

- 4 Edit the **Port** value, and then click **OK**.

Network Pane

Getting there

- From the server console, click **Configuration** > **Network**.

Use the **Network** pane to modify which port or ports the server uses for client access. You can configure the server to listen on multiple ports on the same network adapter or different ports on multiple adapters.

By default the server uses port 22 on all available IPv4 or IPv6 adapters.

Listen Address	Port	Effect
::	22	Listen on port 22 from all available available IPv6 adapters.
0.0.0.0	22	Listen on port 22 from all available IPv4 adapters.

Network Binding Dialog Box

Getting there

- From the server console, click **Configuration** > **Network** > **Add/Edit**.

The options are:

Listening address	<p>Specifies the server IP address. By default the server is set to listen on any available address.</p> <p>Use <code>::</code> to listen to any available IPv6 address.</p> <p>Use <code>0.0.0.0</code> to listen to any available IPv4 address.</p> <p>To configure the server to listen on a specific interface, enter the IP address into this text box. The address must be a valid network address for this computer.</p> <hr/> <p>Note: You need to restart the server after changing this setting.</p>
Port	<p>Specifies the port on which the server listens. The default is 22, which is the standard port for Secure Shell connections.</p> <hr/> <p>Note: You need to restart the server after changing this setting.</p>
Client keep alive	<p>Use this setting to close client sessions that have become unresponsive.</p> <p>When this setting is selected, the server sends a keep-alive packet through the Secure Shell channel to any client that has been idle for the specified number of seconds. If the server receives no response, it closes the client session.</p>
Require reverse DNS lookup	<p>When a client initiates a connection, the server always tries to resolve the client domain name. When Require reverse DNS lookup is selected, the server refuses connections from clients whose domain name cannot be resolved.</p>

The Reflection for Secure IT Data Folder

These items are always stored in the Reflection for Secure IT data folder:

- Configuration file
- Cached credentials

These items are stored by default in the Reflection for Secure IT data folder; you can also configure the server to use a non-default location:

- Host public and private key
- Logs

The data folder location is configurable. The default data folder location is:

Windows Server 2008:

```
\ProgramData\Attachmate\RSecureServer\
```

Windows Server 2003:

```
\Documents and Settings\all users\Application  
Data\Attachmate\RSecureServer\
```

Change the Default Data Folder

Caution: The data folder stores sensitive information, including the host private key and the credential cache. The default data folder is configured to provide access only to SYSTEM and Administrators. This helps secure sensitive data by preventing users from viewing and/or changing the contents of the folder. If you configure an alternate data folder, check the Security settings on that folder, and change these settings, if necessary, so no other users or groups have access to this folder.

To change the default data folder

- 1 From the **Action** menu, select **Set Data Folder**.
- 2 Select **Use custom folder**.
- 3 Use the browse button to locate a folder. The folder must be on the computer running Reflection for Secure IT; network folders are not supported.

- 4 Click **OK** and confirm that you want to change the data folder location. This restarts the server.

Note When you change the **Data folder** setting, Reflection for Secure IT creates a new host key and a new default configuration file in the new location unless these files already exist in the new location. As a result, any settings you have previously configured are no longer used. After you make this change, you can use the console to configure your desired settings and the changes will be saved to the new location. Or, you can copy existing data (such as the host key and your existing configuration file) to the new data folder location.

- 5 Configure server settings using the new data folder location.

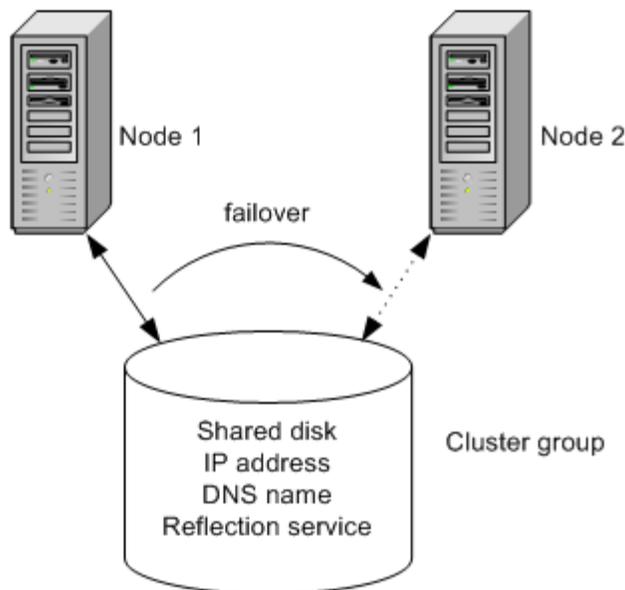
Note: The new data folder location is saved in the Windows registry in the following key location:
HKEY_LOCAL_MACHINE\SOFTWARE\Attachmate\RSecureServer.
This setting remains in the registry if you uninstall or upgrade the server, so subsequent installations continue to use the new location.

CHAPTER 4

Using a Server Cluster

You can configure Reflection for Secure IT to run in a Microsoft cluster environment. The Microsoft cluster service helps ensure that client users have continuous access to your server, even if one computer within the cluster becomes unavailable.

To run in a cluster, you install the Reflection for Secure IT server on multiple nodes, and create a cluster group. This group defines shared resources that can be used by any node in the group. For Reflection for Secure IT, these shared resources include the Reflection for Secure IT service, a shared disk, the server IP address, and the server name. At any given time, only one node has ownership of the shared resources. If that node fails, the Reflection for Secure IT server is started on a different node and that node takes over the shared resources.



In the cluster above, if the Reflection for Secure IT service fails on Node 1, Node 2 acquires the shared resources and the Reflection for Secure IT service is started on the new node. At this point, Node 1 no longer has access to resources within the group. The Reflection for Secure IT server continues to run using the same configuration, so no change is apparent to clients establishing a new connection.

Note: Active client sessions are disconnected when a failover occurs. These clients need to initiate a new session.

Configure a Reflection for Secure IT Cluster

To configure a cluster, you must be running the server in a Microsoft cluster environment. The Microsoft cluster service is required to manage access to shared resources.

Install Reflection for Secure IT

- 1 Install Reflection for Secure IT on each node of your cluster and restart Windows.

Note: Restarting Windows is required to complete the Reflection for Secure IT installation, and this restart automatically starts the Reflection for Secure IT service. In a cluster, the Reflection for Secure IT service should not be started until after the the cluster is correctly configured. The next step ensures that the server is not running on any node until after configuration is complete.

- 2 Launch the Reflection for Secure IT console and stop the server (**Action > Stop Server**).

Configure the cluster

- 1 Open the Microsoft cluster management tool (Failover Cluster Management in Windows 2008 or Cluster Administrator in Windows 2003).
- 2 Create a cluster group for Reflection for Secure IT.
- 3 Add the following items to the Reflection for Secure IT cluster group.

Resource Type	Description
Physical Disk	Location of the Reflection for Secure IT data folder.
IP Address	The IP address used by the server.
Network Name	The host name used by the server.

- 4 Add the Reflection for Secure IT service to the cluster group using the following settings:

Settings	Values
Resource Type	Generic Service
Generic Service Parameters	Set service name equal to: Attachmate Reflection for Secure IT Server Enable this setting: Use network name for computer name
Dependencies	Add the following resources: Physical Disk IP Address Network Name
Registry Replication	Add this HKEY_LOCAL_MACHINE key: SOFTWARE\Attachmate\RSecureServer

- 5 Do this step only if you are running Windows 2008. It ensures that incorrect parameters are not added to the Reflection for Secure IT service startup command.
- On the computer you are using to configure the cluster, open a command window as an administrator. (**Start > All Programs > Accessories**, right-click **Command Prompt > Run as administrator.**)
 - Enter the following command:


```
cluster res "Attachmate Reflection for Secure IT Server" /priv
```
 - If any startup parameters are configured, enter the following to clear the parameters:


```
cluster res "Attachmate Reflection for Secure IT Server" /priv StartupParameters=""
```
 - Repeat step b to verify that there are now no startup parameters configured.

Configure Reflection for Secure IT

- Open the Reflection for Secure IT console on the active node of your cluster group.
- From the **Action** menu, click **Set Data Folder**.
- Select **Enable clustering and use cluster folder**.
- For **Data folder**, specify the folder you want to use for Reflection for Secure IT data. This replaces the *default data folder* (page 205). This should be a local folder on the shared physical disk you have set up as part of your cluster group.

- 5 Configure Reflection for Secure IT server settings.

Note: When you change the **Data folder** setting, Reflection for Secure IT creates a new host key and a new default configuration file in the new location unless these files already exist in the new location. As a result, any settings you have previously configured are no longer used. After you make this change, you can use the console to configure your desired settings and the changes will be saved to the new location. Or, you can copy existing data (such as the host key and your existing configuration file) to the new data folder location.

- 6 Check to be sure that no files or folders configured for use by Reflection for Secure IT reside on any individual node in your cluster. This ensures that files accessed by users will remain available after a failover.

By default, the following settings use the *Windows user profile folder* (page 207) (specified by %D). Depending on your current configuration, you may need to reconfigure the Windows profile folder location, or modify your Reflection for Secure IT settings.

Setting	Notes
User key directory	Used for user public key authentication. The default is %D\.ssh2.
User login directory	The default login directory for SFTP and SCP2 file transfer is %D.

Start Reflection for Secure IT

After the cluster is correctly configured, you can start the service from the Reflection for Secure IT console or from the cluster management tool.

To use	Do this
The Reflection for Secure IT console	Open the console on the active node and start the server (Action > Start Server).
The Microsoft cluster management tool	Bring the Reflection for Secure IT service online.

Set Data Folder Dialog Box

Getting there

- From the **Action** menu, click **Set Data Folder**.

Caution: The data folder stores sensitive information, including the host private key and the credential cache. The default data folder is configured to provide access only to SYSTEM and Administrators. This helps secure sensitive data by preventing users from viewing and/or changing the contents of the folder. If you configure an alternate data folder, check the Security settings on that folder, and change these settings, if necessary, so no other users or groups have access to this folder.

Notes:

- When you change the **Data folder** setting, Reflection for Secure IT creates a new host key and a new default configuration file in the new location unless these files already exist in the new location. As a result, any settings you have previously configured are no longer used. After you make this change, you can use the console to configure your desired settings and the changes will be saved to the new location. Or, you can copy existing data (such as the host key and your existing configuration file) to the new data folder location.
 - Changes you make in this dialog box are saved to the Windows registry in the following key location:
HKEY_LOCAL_MACHINE\SOFTWARE\Attachmate\RSecureServer.
In a cluster environment, these changes are replicated to other nodes in your cluster by the cluster service.
-

The options are:

- **Use default folder** Use the *default data folder* (page 205). Reflection for Secure IT uses this folder to store the host public and private key, the configuration file, the credential cache. It is also the default location for text log files.

Note: To configure a non-default host key or log location, change the values of **Private key** (page 44) or **Log file directory** (page 143) *after* you change your data folder; these settings are saved to the configuration file, which changes when you change the data folder.

Use custom folder Select this option to configure a non-default data folder.

Enable clustering and use cluster folder This option configures Reflection for Secure IT to run in an Microsoft cluster environment. For **Data folder** specify a local folder on the shared physical disk you have set up as part of your cluster group.

Note: To configure a cluster, you must be running the server in a Microsoft cluster environment. The Microsoft cluster service is required to manage access to shared resources.

Data folder Specify the new data folder location. This must be a local folder. Network folders are not allowed. (If **Use default folder** is selected, this option is not available and any path displayed is ignored.)

Note: You need to use the **Browse** button to select the folder. Reflection for Secure IT enforces this in order to ensure that you select an existing folder location.

CHAPTER 5

Data Protection

In this Chapter

Data Encryption	35
Data Integrity	35
Federal Information Processing Standard (FIPS)	36
Encryption Pane	37
Key Exchange	38
Key Exchange Pane	38

Data Encryption

Encryption protects the confidentiality of data in transit. This protection is accomplished by encrypting the data before it is sent using a secret key and cipher. The received data must be decrypted using the same key and cipher. The cipher used for a given session is the cipher highest in the client's order of preference that is also supported by the server.

Reflection for Secure IT Windows Server supports the following data encryption standards:

- Arcfour, Arcfour128, and Arcfour256 (stream mode)
- TripleDES (168-bit) CBC mode
- Cast (128-bit) CBC mode
- Blowfish (128-bit) CBC mode
- AES (also known as Rijndael) (128-, 192-, or 256-bit) CBC mode and CTR mode

Data Integrity

Data integrity ensures that data is not altered in transit.

Secure Shell connections use MACs (message authentication codes) to ensure data integrity. The client and server independently compute a hash for each packet of transferred data. If the message has changed in transit, the hash values are different and the packet is rejected. The MAC used for a given session is the MAC highest in the client's order of preference that is also supported by the server.

Reflection for Secure IT Windows Server supports the following MAC algorithms:

- hmac-sha1
- hmac-sha256
- hmac-sha512
- hmac-md5
- hmac-sha1-96
- hmac-md5-96
- hmac-ripemd160

Federal Information Processing Standard (FIPS)

The United States Government's Federal Information Processing Standard (FIPS) 140-2 specifies security requirements for cryptographic modules. Cryptographic products are validated against a specific set of requirements and tested in 11 categories by independent, U.S. Government-certified testing laboratories. This validation is then submitted to the National Institute of Standards and Technology (NIST), which reviews the validation and issues a certificate. In addition, cryptographic algorithms may also be validated and certified based on other FIPS specifications. The list of certified products and the vendor's stated security policy (the definition of what the module has been certified to do) can be found at: <http://csrc.nist.gov/cryptval/vallists.htm>.

To configure Reflection for Secure IT to run in FIPS mode, select **Use only FIPS-140 certified cryptography algorithms** from the **Encryption** pane and restart the server.

Enabling FIPS Mode has the following effects:

- All connections must be made using algorithms that meet FIPS 140-2 standards. Algorithms that don't meet these standards are not available, except where these algorithms are allowed by NIST for legacy compatibility.
- Minimum public key sizes for both user and host keys and certificates are reset from the default of 512 bits up to 1024 bits. Previously configured keys that do not meet this threshold will not be used.
- If host certificates from the Windows local computer certificate store are used for server authentication, they must have exportable private keys.
- Because Reflection for Secure IT cannot verify the FIPS status of SecurID, GSSAPI/SSPI, and RADIUS binaries, these authentication methods need to be manually disabled by the system administrator if they are not FIPS validated.

Note: You need to restart the server after changing FIPS mode for the change to take effect.

Encryption Pane

Getting there

- From the server console, click **Configuration** > **Encryption**.

The options are:

Ciphers

Specify which ciphers the server allows for encrypting the session. The cipher used for a given session is the cipher highest in the client's order of preference that is also supported by the server.

Caution: Enable None for testing purposes only. When no cipher is used, data is transmitted as clear text.

MACs

Specify which MACs (hashed message authentication codes) the server allows for verifying *data integrity* (page 201). The MAC used for a given session is the MAC highest in the client's order of preference that is also supported by the server.

Caution: Enable None for testing purposes only. When no MAC is used, data is transmitted without integrity checking.

Compression

Specify which compression options the server allows. The options are No Compression or Compression using zlib. The compression used for a given connection is negotiated between the server and client. Compression is desirable on modem lines and other slow connections, but will slow down response rates on fast networks. Compression also adds extra randomness to the packet, making it harder for a malicious person to decrypt the packet.

Use only FIPS-140 certified cryptography algorithms

When selected, allows only those algorithms that meet FIPS 140-2 standards.

Note: You need to restart the server after changing this setting for the change to take effect.

Key Exchange

When a client requests a new session, the server and client use a key exchange protocol to decide on a one-time session key. Key exchange protocols enable the server and client to establish a shared secret key, even though the communications take place in the open. After the key is established, the client and server encrypt subsequent communications using the session key and an agreed upon cipher.

Use the **Key Exchange** pane to control which key exchange protocols the server supports. You can also configure the server to require a new key exchange after a specified time interval has elapsed.

Key Exchange Pane

Getting there

- From the server console, click **Configuration** > **Encryption** > **Key Exchange**.

From this pane, you can enable and disable key exchange algorithms. If you enable only some of the available algorithms, you need to ensure that you select those that are supported by your client(s). The following algorithms are available:

- diffie-hellman-group1-sha1
- diffie-hellman-group14-sha1
- diffie-hellman-gex-sha1
- gss-group1-sha1 with Kerberos 5
- gss-gex-sha1 with Kerberos 5

Secure Shell standards (RFC 4253) require all clients to support both `diffie-hellman-group1-sha1` and `diffie-hellman-group14-sha1`. Of these, `diffie-hellman-group14-sha1` is more secure, but requires more time during the key exchange. `diffie-hellman-gex-sha1` also improves security, and does not slow down the key exchange. However, it is not supported by all clients.

If you use GSSAPI host and user authentication, you need to enable `gss-group1-sha1` and/or `gss-gex-sha1`, depending on your client.

The following option is also available:

**Rekey interval
(seconds)**

Specify the interval (in seconds) after which the server initiates a new key exchange. Setting this value too low can make communication between the client and server impossible. To avoid this problem, it is recommended that you avoid specifying an interval of less than 200 seconds. Use 0 (zero) to turn off rekey requests initiated by the server. Using 0 does not prevent the client from requesting a rekey.

CHAPTER 6

Server Authentication

In this Chapter

Public Key Authentication	42
Configure Public Key Host Authentication	43
Identity Tab	44
Generate Host Private Key Dialog Box	46
Export Public Key Dialog Box	47
Certificate Authentication Overview	47
Configure Certificate Server Authentication	49
Windows Certificate Store Dialog Box	50
Kerberos (GSSAPI) Authentication	51
Configure GSSAPI Server and Client Authentication	52

Authentication is the process of reliably determining the identity of a communicating party. Identity can be proven by something you know (such as a password), something you have (such as a private key or token), or something intrinsic about you (such as a fingerprint).

Secure Shell connections require both server and client authentication.

Server authentication enables the client to confirm the identity of the server. Reflection for Secure IT Windows Server supports these host authentication methods:

- Public key
- Certificate (a special form of public key authentication)
- GSSAPI

The server has only one chance to authenticate to the client during the authentication process. If this authentication fails, the connection fails.

Public Key Authentication

Reflection for Secure IT uses public key server authentication by default. The server automatically generates a new host key (or migrates an existing host key). The default key is an RSA 2048-bit key.

Public key cryptography uses a mathematical algorithm with a public/private key pair to encrypt and decrypt data. One of the keys is a public key, which can be freely distributed to communicating parties, and the other is a private key, which should be kept secure by the owner of the key. Data encrypted with the private key can be decrypted only with the public key; and data encrypted with the public key can be decrypted only with the private key.

When keys are used for authentication, the party being authenticated creates a digital signature using the private key of a public/private key pair. The recipient must use the corresponding public key to verify the authenticity of the digital signature. This means that the recipient must have a copy of the other party's public key and trust in the authenticity of that key.

How it Works

When public key authentication is used for host authentication, the following sequence of events takes place.

1. The Secure Shell client initiates a connection.
2. The server sends its public key to the client.
3. The client looks for this key in its trusted host key store.

If the client	This occurs
Finds the host key, and the client copy matches the key sent by the server	Authentication proceeds to the next step.
Does <i>not</i> find the host key	The client displays a message that the host is unknown and provides a fingerprint of the host key. If the client is configured to allow the user to accept unknown keys (the default), the user can accept the key, and authentication proceeds to the next step. If strict host key checking is enforced, the client ends the connection.
Finds a host key, and the client copy <i>doesn't</i> match the key sent by the server	The client displays a warning that the key doesn't match the existing key and displays the fingerprint of the key sent by the server. If the client is configured to allow the user to accept unknown keys (the default), the user can accept the new key. If strict host key checking is enforced, the client ends the connection.

4. To confirm that the server actually holds the private key that corresponds to the received public key, the client sends a challenge (an arbitrary message) to the server and computes a *hash* (page 202) based on this message text.
5. The server creates a digital signature based on the challenge message. To do this, the server independently computes the message hash, and then encrypts the computed hash using its private key. The server attaches this digital signature to the original challenge and returns this signed message to the client.
6. The client decrypts the signature using the public key and compares the hash with its own computed hash. If the values match, host authentication is successful.

Configure Public Key Host Authentication

The server is configured to use host public key authentication by default. This means that client users see an unknown host key message the first time that they connect to the server. You may want to export the host public key and install it on client computers, so that client users can connect without having to verify the server identity.

To configure public key authentication on the server

- 1 Start the server console, and then click **Identity**.

To	Do this
Use the default key	<i>No action required.</i>
Generate a new key	Click Generate .
Use a different key	Click Browse .

- 2 Save your settings (**File > Save Settings**).

To export the host public key

- 1 Start the server console, and then click **Identity**.
- 2 Under **Host key**, click **Export**.
- 3 Specify name and location for the exported key, and then click **Save**.

Refer to your client documentation for information about how to add the exported key to the client's known hosts list. The procedure that follows is for the Reflection for Secure IT Windows Client.

To add the server key to the client known hosts list

- 1 Copy the exported public key file to a location on or available to the client computer. (The key doesn't need to remain here after the import is complete.)
- 2 Start Reflection for Secure IT Windows Client.
- 3 Open the **Reflection Secure Shell Settings** dialog box (**Connection > Connection Setup > Security**).
- 4 From the **Host Keys** tab, click **Import**.

Identity Tab

Getting there

- From the server console, click the **Identity** tab.

Use the **Identity** tab to configure server authentication.

Host Key

Private key	Specifies the filename and location of the private key used to authenticate the server.
Key comment	Displays comment text, which includes identifying information about the key.
SHA1 fingerprint	Displays the SHA1 <i>hash</i> (page 202) for this key. Use this value to confirm the host identity when a client displays an unknown host fingerprint using SHA1 (also called Bubble Babble) format.
MD5 fingerprint	Displays the MD5 hash for this key. This is the hexadecimal value of the public key. Use this value to confirm the host identity when a client displays an unknown host fingerprint using MD5 format.
Generate	Opens the Generate Host Private Key dialog box, from which you can create a new host key.
Export	Uses the host private key to create the associated public key. You can add the exported key to a client's trusted host store.

Use host certificate

Use host certificate

When this option is cleared (the default), the server always authenticates using its public key.

When this option is selected, the server can authenticate using either its public key or a host certificate. (The authentication method used depends on the client configuration).

Use the local computer certificate from the Windows certificate store

Select this option to use a local computer certificate from the Windows certificate store.

Click **Browse** to select a certificate from this store.

Click **View** to view the contents of the selected certificate.

Use the following certificate

Select this option to authenticate using a certificate in a file available on your system. You can use this option with either of the following:

- A PKCS #12 file (*.pfx or *.p12) that includes both the certificate and the associated private key.
- or-
- A certificate file (*.cer) and its associated private key.

Private key

Specify the filename and location of a private key, or a PKCS#12 file that includes the private key.

Note: The private key used for host authentication cannot be passphrase-protected.

Certificate

Specify the name and location of the certificate.

- If you specify a PKCS#12 file for **Private key**, the certificate is automatically exported, and the correct name and location are entered automatically.
- If the client is not configured for certificate authentication, the server uses public key authentication, even if you have configured certificate authentication on the server.

Server version string

Server version string

A two-part string sent to the client when a connection is made.

The first part of the string (SSH-2.0-) consists of the SSH version supported by the server, and cannot be edited. The second portion of the string is handled as follows:

If you	This occurs
Do not edit this string	The value is generated automatically, and includes the server's build number. This number will be updated automatically when you upgrade your server software.
Edit this string	The edited value is saved to your configuration file, and your edited string is not affected by subsequent software upgrades.

Note: This value is not saved in the configuration file.

Note: Many Secure Shell clients use the server version string to identify the server manufacturer and modify client behavior to match the server type. If you edit this string, users may encounter unexpected client functionality.

Generate Host Private Key Dialog Box

Getting there

- From the server console, click **Identity** > **Generate**.

From this dialog box, you can generate a new host key.

Note: Generating a new key automatically updates the value for **Private key** on the **Identity** tab. Use **File** > **Save** to save this change to your configuration file.

Host key

Specifies the filename for the generated private key. (A public key is also created and is always given the same name as the private key plus a .pub file extension.) To edit the filename or change the folder location, click **Browse**.

Key size	<p>Specifies the key size. Up to a point, a larger key size improves security. Increasing key size slows down the initial connection, but has no effect on the speed of encryption or decryption of the data stream after a successful connection has been made. The length of key you should use depends on many factors, including: the key type, the lifetime of the key, the value of the data being protected, the resources available to a potential attacker, and the size of the symmetric key you use in conjunction with this asymmetric key. To ensure the best choice for your needs, we recommend that you contact your security officer.</p> <p>The list of available sizes includes commonly used key sizes. If you need a key with a different size, you can use the ssh-keygen <i>command line utility</i> (page 193). (Key sizes specified with this utility are rounded up to the next value evenly divisible by 64 bits.)</p>
Key type	Specifies the algorithm used for key generation.

Export Public Key Dialog Box

Getting there

- From the server console, click **Identity** > **Export**.

Use this dialog box to specify the filename, location, and format of the exported public key derived from the host private key.

Host key	Specifies the filename to be used for the exported public key. To specify a filename or change the folder location, click Browse .
SECSH format	The key format used for storing public keys on Reflection for Secure IT, F-Secure, and SSH Communications clients.
OpenSSH format	The key format used for storing public keys on OpenSSH clients.

Certificate Authentication Overview

Reflection for Secure IT supports host (server) authentication using certificates.

Certificate authentication solves some of the problems presented by public key authentication. With public key host authentication, the system administrator must either add the host public key for every server to each client's list of known hosts, or count on client users to confirm the host identity correctly when they connect to an unknown host. Certificate authentication avoids this problem by using a trusted third party, called the certification authority (CA), to verify the validity of information coming from the host.

Like public key authentication, certificate authentication uses public/private key pairs to verify the host identity. However, with certificate authentication, public keys are contained within digital certificates, and in this case, two key pairs are used; the host holds one private key and the CA holds a second. The host obtains a certificate from the CA. This certificate contains identifying information about the host, a copy of the host public key, and a digital signature created using the CA's private key. This certificate is sent to the client during the authentication process. To verify the integrity of the information coming from the host, the client must have a copy of the CA's public key, which is contained in the CA root certificate.

Installing CA root certificates to verify the host identity has several advantages over installing and configuring host public keys:

- A single CA certificate can be used to authenticate multiple servers.
- Administrators can use Windows Group Policies to install CA certificates on Windows clients.
- Root certificates for commercially obtained certificates may already be available on client computers.
- If necessary, the host can obtain a new certificate from the same CA with no change required on client systems.

How it Works

Server certificate authentication uses the following sequence of events:

1. The Secure Shell client initiates a connection.
2. The host sends its certificate to the client.
3. The client uses the CA root certificate to verify the validity of the server certificate.

Note: The client must already have a copy of the CA certificate in the trusted root store. (A single CA certificate can be used to authenticate multiple servers.)

4. The client checks that the server information in the host's certificate matches the host being contacted.
5. To confirm that the host holds the private key that corresponds to the public key in the certificate, the client sends a challenge (an arbitrary message) to the server and computes a *hash* (page 202) based on this message text.
6. The server creates a digital signature based on the challenge message — the server independently computes the message hash, and then encrypts the computed hash using its private key. Next, the server attaches this digital signature to the challenge and returns this signed message to the client.
7. The client decrypts the signature using the server's public key and compares the hash with its own computed hash. If the values match, host authentication is successful.

Configure Certificate Server Authentication

You can configure the server to authenticate using any of the following:

- The local computer certificate stored within the Windows certificate store.
- A PKCS #12 file (*.pfx or *.p12) that includes both the certificate and the associated private key.
- A certificate file (*.cer) and its associated private key.

Here's a quick summary of the important steps. The details are explained in the procedures that follow.

1. Configure the server for certificate authentication.
2. Install the CA root certificate on the client.
3. (Optional) Configure strict host key checking on the client.

To configure certificate authentication on the Reflection for Secure IT server

- 1 Start the server console, and then click **Identity**.
- 2 Select **Use host certificate** and specify the certificate to use.

To use	Do this
The local computer certificate from the Windows store	Select Use the local computer certificate from the Windows certificate store . Click Browse to select a certificate from this store.

To use	Do this
A certificate in a PKCS#12 file	Select Use the following certificate , and then in the Private key text box, enter the full path and filename (*.pfx or *.p12). The certificate is exported automatically, and the exported file appears in the Certificate text box.
A certificate and its associated private key	Select Use the following certificate , enter the full path and name of the private key file in the Private key text box, and then specify the full path and name of the certificate file in the Certificate text box.

3 Save your settings (**File** > **Save Settings**).

4 Restart the server.

The procedure that follows describes how to configure the Reflection for Secure IT Windows Client for server certificate authentication. If you use a different client, refer to your client documentation.

To configure the Reflection for Secure IT Windows Client

- 1 Start Reflection for Secure IT Windows Client.
- 2 Open the **Reflection Secure Shell Settings** dialog box (**Connection** > **Connection Setup** > **Security**).
- 3 Click the **PKI** tab.
- 4 Install the CA root certificate on the client:

To add the certificate to	Do this
The Windows certificate store	Click View System Certificates , and then import the certificate using the Trusted Root Certification Authorities tab.
The Reflection certificate store	Click Reflection Certificate Manager , and then import the certificate using the Trusted Root Certification Authorities tab.

- 5 (Optional) To eliminate the risk created by allowing users to accept unknown keys, enforce strict host key checking on the client — from the **Host Keys** tab of the **Secure Shell Settings** dialog box, set **Enforce strict host key checking** to Yes.

Windows Certificate Store Dialog Box

Getting there

- 1 From the server console, click the **Identity** tab.
- 2 Select **Use the local computer certificate from the Windows certificate store**.
- 3 Click **Browse**.

This dialog box lists local computer certificates available in the Windows certificate store. To configure the server to authenticate using a certificate, select the certificate and click **OK**.

Kerberos (GSSAPI) Authentication

Kerberos is a security protocol that provides an alternate mechanism for both client and server authentication. Kerberos authentication relies on a trusted third party called the KDC (Key Distribution Center). The Secure Shell protocol supports Kerberos authentication via GSSAPI (Generic Security Services Application Programming Interface).

Reflection for Secure IT supports Kerberos authentication when the KDC is a Windows domain controller. Both the client user and server host must be part of the same Windows domain.

Note: Windows operating systems starting with Windows 2000 manage authentication using Kerberos version 5. The KDC is maintained on the Windows domain controller and Active Directory is used to manage the security account database.

Advantages of using Kerberos authentication include:

- Using a trusted third party eliminates the key management tasks you encounter when you use public key authentication.
- Client users who log into the Windows domain need no additional authentication to connect to the Reflection for Secure IT server.
- When Kerberos is used for server authentication, no host key is required. This means that client users won't need to respond to an unknown host prompt.

Server Authentication using GSSAPI

By default, Secure Shell connections are established using this sequence of events:

1. Key exchange — the client and server negotiate a shared secret key, cipher, and hash for the session.
2. Server authentication — by default, the server presents a host key for this purpose.
3. Client authentication.

When GSSAPI is used for server authentication, the Kerberos KDC authenticates the server during the initial key exchange. No subsequent server authentication is needed, and the server never sends a host key to the client.

Client Authentication using GSSAPI

After a user has authenticated to a Windows domain, that user holds Kerberos credentials that can be used by other Kerberized applications. When you configure Reflection for Secure IT to support GSSAPI, the server uses Kerberos credentials to authenticate client users. This means that users who have authenticated to the Windows domain need no additional authentication to connect to the server.

Configure GSSAPI Server and Client Authentication

If the server host computer and client users are members of the same Windows domain, you can use GSSAPI for mutual authentication. With this configuration, both the client and server authenticate using Windows domain credentials. No host key is required and the user needs no password to connect to the server.

Note: The following procedures enable both client and server authentication using Windows domain credentials. Configuring just client authentication requires fewer steps. If you don't need GSSAPI for server authentication, see *Configure Client Authentication using Windows Credentials* (page [77](#)).

To configure Windows domain accounts

- 1 Add the server computer and client computers to the Windows domain.
- 2 Launch the Active Directory Users and Computers console and add the client users to the domain.
- 3 Configure the user accounts to use DES encryption. (**Account** > **Account options** > **Use Des encryption types for this account**).

Note: This change is required by the Reflection Kerberos client, and is needed only for GSSAPI server authentication.

- 4 (Optional) If you want to use delegation of authentication, configure user account to be trusted for delegation (**Account** > **Account options** > **User is Trusted for delegation**).
- 5 (Optional) If you want to use delegation of authentication, configure the server computer properties to trust this computer for delegation (**General** > **Trust computer for delegation**).

To configure the Reflection for Secure IT server

- 1 Start the server console, and then click **Configuration**.
- 2 Go to **Authentication** > **GSSAPI / Kerberos V5**, and then select **Allow** or **Require**.
- 3 Go to **Encryption** > **Key Exchange** and confirm that the following (default) key exchange protocols are selected: **gss-group1-sha1 with Kerberos 5** and **gss-gex-sha1 with Kerberos 5**.

To configure the Reflection for Secure IT Windows Client

- 1 Start Reflection for Secure IT Windows Client.
- 2 Open the **Reflection Secure Shell Settings** dialog box (**Connection** > **Connection Setup** > **Security**).
- 3 From the **General** tab, under **Authentication**, select **GSSAPI/Kerberos**.
- 4 From the **GSSAPI** tab, select **Reflection Kerberos** and click **Configure**.
- 5 Configure Reflection Kerberos to use Windows logon credentials.

If	Do this
This is the first time you've used Reflection Kerberos	Click Use Windows logon values in the Reflection Kerberos Initial Configuration dialog box. The Realm and KDC host values are supplied automatically.
You have already configured Reflection Kerberos	Set your Windows domain as your default realm and configure it to use Windows logon credentials. (Configuration > Configure Realms > Properties > Use Windows logon credentials).

- 6 Use the Reflection Kerberos Manager to remove DES3_HMAC_SHA1 from the list of requested KDC encryption types. To edit this list use **Configuration** > **Configure Realms** > **Properties** > **Encryption** > **Configure Encryption Types**.

- 7 When you configure the user for your client connection, you may need to include both the domain and user name using the format `domain\user`. This is required if the server computer has a local account name that matches your domain account. For example, if the local computer has a "joe" account and you log on using a domain account for "joe", you need to connect from the client as:

```
mydomain\joe
```

Note: Depending on your operating system, you may need to modify your system security settings to allow access to a terminal shell to users who authenticate using domain credentials. For more information, see [Command Shell Access in Windows](#) (page [124](#)).

CHAPTER 7

User Authentication

In this Chapter

Authentication Pane	55
Password and Keyboard Interactive Authentication	57
Public Key Authentication for Users	61
Certificate Authentication for Users	64
RSA SecurID Authentication	72
RADIUS Authentication	74
GSSAPI (Kerberos) Authentication	77

Authentication is the process of reliably determining the identity of a communicating party. Identity can be proven by something you know (such as a password), something you have (such as a private key or token), or something intrinsic about you (such as a fingerprint).

Secure Shell connections require both server and client authentication. Several methods of client authentication are available, and both the client and server can be configured to determine which method — or methods — are used. The server can be configured to allow, require, or deny client authentication methods. During Secure Shell connection negotiations, the server presents a list of allowed and required methods from which the client and server negotiate one or more authentication methods.

Authentication attempts follow the order of preference set by the client. The connection uses the first authentication technique highest in the client order of preference that is also allowed by the server. If the server is configured to require more than one method, multiple authentication methods are needed to establish a connection.

Authentication Pane

Getting there

- From the server console, click **Configuration** > **Authentication**.

The options are:

Login grace time

Grace time for completion of authentication process (seconds)

Sets the number of seconds allowed for client authentication. If the client fails to authenticate the user within the specified number of seconds, the server disconnects and exits. Use zero (0) to set no limit.

Note: Specifying no limit (0) is not recommended. Unauthenticated connections use up system resources and can lead to a denial-of-service condition.

IP blocking

You can use the **IP blocking** settings to temporarily block connections from any client IP address that has exceeded a specified number of failed attempts. If a particular IP address exceeds the value set for **Failed attempts**, within the time period specified by **Failure time-out**, that IP address is blocked for the duration specified by **Lockout duration**.

Notes:

- IP blocking applies only to password authentication (both traditional password and password over Keyboard Interactive).
 - You can disable the IP Blocking feature by setting **Failed Attempts** to 0 (zero).
 - IP blocking information is stored in memory, and is cleared if the server is restarted.
 - You can lock out offending addresses permanently from the **Client Host Access Control** pane.
-

Failed attempts

Sets a maximum number of failed login attempts. To disable IP blocking, set this value to zero (0). The default is 20.

Failure time-out (seconds)

Sets a duration of time, in seconds, during which an IP address is monitored for failed login attempts. The default is 300 seconds (5 minutes).

Lockout duration (seconds) Sets the number of seconds an IP address remains blocked after the value set for **Failed attempts** is exceeded. The default is 3600 seconds (one hour).

Authentication failures

You can use the Authentication failures settings to modify how the server handles client authentication failures.

Caution: Enabling the the settings in the **Authentication failures** group increases your security risk by providing potential attackers with information about which client accounts are valid.

Immediately disconnect invalid, locked or denied users By default, this option is not selected, and the server responds identically to all failed authentication attempts. When this setting is selected, blocked accounts disconnect immediately.

Provide informative messages to clients for authentication failures By default, no information about authentication failures is sent to the client. This complies with SSH convention. When this setting is selected, the client is told when an authentication fails because of a invalid, locked, or denied user account.

Keyboard interactive

Send keyboard interactive title This setting affects whether or not title text is displayed during keyboard interactive authentication. When it is selected (the default), the authentication prompts sent to the client include the title text. When it is not selected, the title text is not included.

Password and Keyboard Interactive Authentication

Reflection for Secure IT server supports both password and keyboard interactive authentication by default.

Authentication method	Description
Password	<p>Prompts the client user for the login password for that user on the Secure Shell server host.</p> <p>The password is sent to the host through the encrypted channel.</p>
Keyboard interactive	<p>Supports any procedure in which authentication data is entered using the keyboard, including simple password authentication, thereby enabling the Secure Shell client to support a range of authentication mechanisms, such as RSA SecurID tokens or RADIUS servers.</p> <p>A client administrator could, for example, configure keyboard interactive authentication to handle situations in which multiple prompts are required, such as for password updates.</p> <p>Keyboard data is sent to the host through the encrypted channel.</p> <hr/> <p>Note: Configure keyboard interactive authentication from the Password pane.</p>

Configure Password User Authentication

Use the **Password** pane to configure password authentication settings. Password authentication must also be supported by the client.

To configure password authentication on the server

- 1 Start the server console, and then click **Configuration**.
- 2 Go to **Authentication > Password**.
- 3 Configure the server to allow, require, or deny password authentication.
 - When you select **Deny**, both password authentication and keyboard interactive authentication are denied, regardless of whether **Password authentication using keyboard interactive** is selected.
 - When you select **Require** and **Password authentication using keyboard interactive** is not selected, the server requires traditional password authentication; keyboard interactive authentication is not required.

- 4 Configure additional password authentication options, including:
 - The number of password attempts to allow, and the delay between tries.
 - Whether to permit empty passwords, and/or whether to allow password changes.

Note: When **Password authentication using keyboard interactive** is not selected, these settings apply to traditional password authentication only; when selected, these settings also apply to keyboard interactive authentication.

- 5 Save your settings (**File > Save Settings**).

Configure Keyboard Interactive User Authentication

Use the **Password** pane to configure keyboard interactive authentication. Keyboard interactive authentication must also be supported by the client.

To configure keyboard interactive authentication on the server

- 1 Start the server console, and then click **Configuration**.
- 2 Go to **Authentication > Password**.
- 3 Select **Password authentication using keyboard interactive** (the default).
- 4 Configure the server to allow, require, or deny password authentication.
 - When you select **Deny**, both password authentication and keyboard interactive authentication are denied, regardless of whether **Password authentication using keyboard interactive** is selected.
 - When you select **Require** and **Password authentication using keyboard interactive** is selected, the server requires keyboard interactive authentication; traditional password authentication is not required.
- 5 Configure additional password authentication options, including:
 - The number of password attempts to allow, and the delay between tries.
 - Whether to permit empty passwords, and/or whether to allow password changes.

Note: When **Password authentication using keyboard interactive** is not selected, these settings apply to traditional password authentication only; when selected, these settings also apply to keyboard interactive authentication.

- 6 Save your settings (**File > Save Settings**).

Password Pane

Getting there

- From the server console, click **Configuration** > **Authentication** > **Password**.

From the password pane, you can configure both traditional password authentication and keyboard interactive authentication.

Note: Items on this pane can be configured globally or as part of a *subconfiguration* (page [129](#)).

Password authentication

Allow	<p>When Password authentication using keyboard interactive is not selected, the server allows only traditional password authentication.</p> <p>When Password authentication using keyboard interactive is selected, the server allows both traditional password authentication and keyboard interactive authentication.</p>
Require	<p>When Password authentication using keyboard interactive is not selected, the server requires traditional password authentication.</p> <p>When Password authentication using keyboard interactive is selected, the server requires keyboard interactive authentication.</p>
Deny	<p>Denies both password authentication and keyboard interactive authentication, regardless of whether Password authentication using keyboard interactive is selected.</p>

Note: When **Password authentication using keyboard interactive** is not selected, the settings below apply only to traditional password authentication; when it is selected, these settings also apply to keyboard interactive authentication.

Number of password attempts	<p>If the client is configured to allow a larger number of attempts, the client user sees the larger number of prompts, but the value specified here sets the actual limit.</p> <p>If the client is configured to allow a smaller number of attempts, the client sets the actual limit.</p>
Delay between tries (seconds)	<p>Sets the number of seconds the server should wait to send between prompts.</p>
Permit empty passwords	<p>Empty passwords must also be supported in your Windows Group Policy configuration.</p>

Allow password change	Specifies whether users are allowed to change their password.
------------------------------	---

Public Key Authentication for Users

Public key authentication relies upon public/private key pairs. To configure public key authentication, each client user needs to create a key pair and upload the public key to the server. If the key is protected by a passphrase, the client user is prompted to enter that passphrase to complete the connection using public key authentication.

Configure Public Key User Authentication: Reflection for Secure IT Windows Clients

Public key authentication requires both client and server configuration. The Reflection for Secure IT Windows Client includes an upload utility that simplifies this process.

Note: If you have not changed the default public key settings, you don't need to make any changes to the server settings.

To configure the server

- 1 Start the server console, and then click **Configuration**.
- 2 Go to **Authentication > Public Key** and select **Allow** (the default). If you want to require public key authentication, don't make this change until after you upload the client keys.
- 3 Use the default values for **User key directory** and **Authorization file name**.
- 4 Save your settings (**File > Save Settings**).

The next procedure generates the key pair on the client, uploads the public key to the server, and automatically updates the client user's authorization file.

To configure the Reflection for Secure IT Windows Client

- 1 Start Reflection for Secure IT Windows Client.
- 2 Open the **Reflection Secure Shell Settings** dialog box (**Connection > Connection Setup > Security**).
- 3 From the **General** tab, under **User authentication**, confirm that **Public key** is selected.
- 4 From the **User Keys** tab, click **Generate Key**, specify your key options, and then click **Create**.

- 5 Select the key in your user key list and click **Upload**.
- 6 Enter your host, user name, and password in response to the prompts.
- 7 Click **OK** in response to the key destination prompt.

The default destination and authorization filenames in the destination prompt match the Reflection for Secure IT server defaults.

Note: Public key authentication is not supported for the local Guest account.

Configure Public Key User Authentication: Reflection for Secure IT UNIX Clients

Public key authentication for Reflection for Secure IT UNIX clients requires both client and server configuration. Here's a quick overview of the main steps involved. The details are explained in the procedures that follow.

1. Create a key pair on the client.
2. Add a line to the client identification file (`~/.ssh2/identification`) that identifies the private key.
3. Copy the public key to the user's directory on the server (`~/.ssh2`).
4. Add a line to the user's authorization file (`~/.ssh2/authorization`) on the server that identifies the public key.

To configure the Reflection for Secure IT UNIX Client

- 1 Generate a public/private key pair using the **ssh-keygen** utility. For example:

```
ssh-keygen mykey
```

- 2 In a text editor, open (or create) the client identification file. The default name and location for this file is `~/.ssh2/identification`.
- 3 Add a line to the client identification file that identifies the private key you created (using the format `"IdKey"` for the key entry, followed by the name of the private key file). For example:

```
IdKey /home/joe/mykey
```

To configure public key user authentication on the Windows server

- 1 Start the server console, and then click **Configuration**.
- 2 Go to **Authentication** > **Public Key** and select either **Allow** or **Require**.

- 3 (Optional) From the **User key directory** box, specify a folder in which to store user public keys. By default, the server looks for keys in an `.ssh2` subfolder in the *user folder* (page [207](#)).

Note: This folder must be SFTP-accessible.

- 4 (Optional) Specify a filename for the authorization file. By default, the server uses the name `authorization`.
- 5 Copy the client public key to the user key directory on the server. For example, the default location for joe on Windows Server 2008 would be:

```
C:\Users\joe\.ssh2
```

Note: To create a folder with a name starting with a dot, you need to use the DOS command window.

- 6 Using a text editor, create or edit the authorization file for this user. For example, the default file for joe on Windows Server 2003 would be:

```
C:\Users\joe\.ssh2\authorization
```

- 7 Add a line to the authorization file that identifies the key you copied to the user key directory (using the format `"key"` for key entries, followed by the public key name). For example:

```
key mykey.pub
```

Note: Public key authentication is not supported for the local Guest account.

Public Key Pane

Getting there

- From the server console, click **Configuration** > **Authentication** > **Public Key**.

Use the **Public Key** pane to configure user authentication using public keys.

Note: Items on this pane can be configured globally or as part of a *subconfiguration* (page [129](#)).

Public key authentication

Allow/Require/Deny Allow is the default.

Public key storage

User key directory Specifies the directory used for storing user public keys on the server. You can specify any physical directory, or use one of the following pattern strings to specify user-specific directories. For details see *Pattern Strings in Directory Paths* (page [165](#)).

%D The user's *User profile folder* (page [207](#)).

%H The user's *Home folder* (page [206](#)).

%u The user's login name.

%U The user's domain name and login in the format `domain.username`.

Note: Do not use %u or %U to point to a location within a user's Windows profile folder. Neither of these options works correctly for this purpose. Use these options to create your own user-specific locations in some other location, for example on a shared network file server. For details, see *Pattern Strings in Directory Paths* (page [165](#)).

Authorization file name The name of the file in the user key directory that contains a list of public keys that can be used for user authentication. The default name is `authorization`.

Size

Public key minimum length (bits) Specifies the minimum allowable key size. The default is 512. Allowed values are 512-8192.

Public key maximum length (bits) Specifies the maximum allowable key size. The default is 8192. Allowed values are 512-8192.

Certificate Authentication for Users

Using certificates for client authentication solves some of the problems presented by public key authentication. With public key authentication, each client must upload a copy of the public key to every server. Certificate authentication avoids this problem by using a trusted third party, the certification authority (CA), to verify the validity of information coming from the client. With certificates, you can configure authentication using a single trust anchor instead of multiple unique client public keys.

Note: Reflection PKI Services Manager supports central management of PKI settings. You can install and configure a single instance of PKI Services Manager to provide certificate validation services for all supported Attachmate products.

Requirements

Requirement	Function
Reflection PKI Services Manager must be installed and correctly configured.	PKI Services Manager validates the certificate and uses a map file to determine which users can authenticate with a valid certificate. You need to configure at least one trust anchor and one mapping rule for certificate validation to succeed. You may also need to configure access to intermediate certificates and to certificate revocation information.
A certificate signed by a CA and the associated private key must be installed on the client.	The client sends this certificate to the server to authenticate the user.
The Reflection for Secure IT server must have a copy of the PKI Services Manager public key and be configured to connect to PKI Services Manager.	The server communicates with PKI Services Manager to confirm the validity of the user certificate.

Using PKI Services Manager

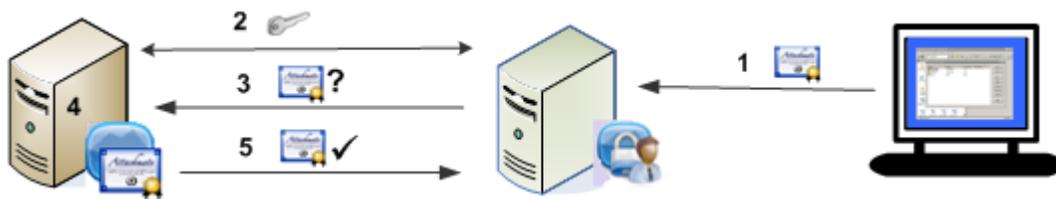
Reflection PKI Services Manager is a service that provides X.509 certificate validation services. If your client users configure authentication using certificates, you need to download and install this application. It is available at no additional charge.

- Reflection PKI Services Manager is required by the server to authenticate clients that use certificates.
- Reflection PKI Services Manager is supported on both Windows and UNIX platforms.
- Reflection PKI Services Manager supports central management of PKI settings. You can install and configure a single instance of PKI Services Manager to provide certificate validation services for all supported Attachmate products.

This user guide provides basic information about installing PKI Services Manager and configuring Reflection for Secure IT to use it for certificate validation services. For additional information, refer to the PKI Services Manager documentation at <http://support.attachmate.com/manuals/pki.html>.

How it Works

- 1 The Secure Shell client presents a certificate to the server for user authentication.
- 2 The Reflection for Secure IT server connects to Reflection PKI Services Manager and verifies its identity using an installed public key.
- 3 Reflection for Secure IT sends the certificate and the user name to PKI Services Manager.
- 4 PKI Services Manager determines if the certificate is valid and whether the user is allowed to authenticate with this certificate based on the rules the PKI Services Manager administrator has configured on the PKI Services Manager Identity Mapper pane. This information is returned to Reflection for Secure IT.
- 5 If the certificate is valid and the user presenting it is an allowed identity for this certificate, Reflection for Secure IT validates the user's digital signature. If the digital signature is verified, user authentication is successful.



Configure Certificate Authentication for Users

Before you begin, review the requirements described in the *Certificate Authentication for Users* (page [64](#)) topic.

To configure user authentication using certificates, you need to install and configure Reflection PKI Services Manager, and configure your server and client. Use the following procedures to get started. Many additional variations are possible. For more information, see the Reflection PKI Services Manager User Guide, which is available from <http://support.attachmate.com/manuals/pki.html>.

You can install and configure a single instance of PKI Services Manager to support user certificate authentication for multiple Reflection for Secure IT servers. However, because Reflection for Secure IT settings allow only one entry for the PKI Services Manager address and port, this configuration creates a potential single point of failure. If PKI Services Manager is unreachable or the server is not running, all authentication attempts using certificates will fail. In order to provide load balancing and failover, you can define a round-robin DNS entry for the PKI Services Manager host name or place the PKI Services Manager host behind a load balancing server.

To install and configure PKI Services Manager

Note: You can configure PKI Services Manager using the console, or by editing the configuration files directly. Default settings and map files are created in the *data folder* (page [204](#)) the first time you run the console and are updated when you make and save changes using the console. These instructions use the console.

- 1 *Install Reflection PKI Services Manager* (page [12](#)).
- 2 Start the PKI Services Manager console:
Programs > Attachmate Reflection > Utilities > PKI Manager
- 3 Put a copy of the certificate you want to designate as a trust anchor into your local store. The default PKI Services Manager store is in the following location:

```
common application data folder\Attachmate\ReflectionPKI\local-store
```

(This step is not required if you are using certificates in the Windows store or you have a copy of the trust anchor available somewhere else on your system.)

- 4 From the **Trusted Chain** pane, add your trust anchor (or anchors) to the list of trust anchors.

To use this store	Do this
Your local certificate store or a certificate file on your system	Click Add . Select either Local store certificate or Certificate file , click Browse and select the certificate for your trust anchor.
The Windows certificate store	Under Search order to use when building path to trust anchor , select "Windows certificate store." Click Add . From the Add Trust Anchor dialog box, select Windows certificate then click Browse to select an available certificate.

Note: PKI Services Manager uses only those certificates that are installed for use by the local computer (not certificates installed for the current user) and are in either the trusted root certification authorities list or the trusted intermediate authorities list. To view and manage the local computer certificates, use the Microsoft Management Console. Add the Certificates Snap-in and configure it to manage certificates for the computer account.

- 5 From the **Revocation** pane, configure certificate revocation checking.

Note: By default PKI Services Manager looks for CRLs in the local store. If you use this configuration, you need to copy the CRLs to your local store.

- 6 From the **Identity Mapper** pane, add rules to determine which identities can authenticate with a valid certificate.

After a certificate is determined to be valid, rules are processed in order (based on rule type then sequence). If the certificate meets the requirements defined in the conditional expression (or if the rule has no condition), the allowed identities specified in that rule are allowed to authenticate. No additional rules are applied after the first match.

Note: When you create rules for authentication of Windows domain users, you need to include both the domain and user name in this format:
domain\user.

- 7 Click **File > Save**.
- 8 *Start the PKI Services Manager service* (page 71) if it isn't already running. If the service is already running, reload your settings (**Server > Reload**).

To configure the Reflection for Secure IT server

- 1 If PKI Services Manager is not installed on the same host as the Reflection for Secure IT server, copy the PKI Services Manager public key to the Reflection for Secure IT server. The key location on PKI Services Manager is:

```
common application data
folder\Attachmate\ReflectionPKI\config\pki_key.pub
```

Copy this to any location on the Reflection for Secure IT host. For example:

```
\ProgramData\Attachmate\RSecureServer\pki_key.pub
```

-or-

```
C:\Documents and Settings\All Users\Application
Data\Attachmate\RSecureServer\pki_key.pub
```

- 2 Start the Reflection for Secure IT console.
- 3 From the **Public Key** pane, ensure that **Public key authentication** is set to **Allow** or **Require**. (**Allow** is the default.)
- 4 From the **Certificates pane**, edit **PKI server** and **Port** to match the **PKI server address** values configured in the PKI Services Manager console.
- 5 For **Public key file**, specify the location to which you copied the key in step 1.
- 6 Save your settings (**File > Save Settings**).

To configure the client

You will need to perform the following basic steps on your Secure Shell client computer. Reflection for Secure IT Windows client procedures are included here. For other clients, refer to the client documentation.

- 1 Install the user certificate and associated private key on the client computer.

For connections from the Reflection for Secure IT Windows client, you can import certificates using PKCS#12 files (typically *.pfx or *.p12) that contain a certificate and its associated private key. You can import these to either the Windows certificate store or the Reflection certificate store. Access to both stores is available from the **Reflection Secure Shell Settings** dialog box using buttons on the **PKI** tab.

- 2 Configure the client to authenticate using the certificate.

In the Reflection for Secure IT Windows client, open **Reflection Secure Shell Settings** dialog box, and select the **User Keys** tab. Certificates you have imported into the Windows and Reflection stores are automatically included in the list of available keys. Select the certificate(s) you want to use for authentication.

- 3 Confirm that the client supports public key authentication.

All Reflection clients support public key authentication by default. To confirm authentication settings from the **Reflection Secure Shell Settings** dialog box, go to the **General** tab.

Certificates Pane

Getting there

- From the server console, click **Configuration** > **Authentication** > **Public Key** > **Certificates**.

Configure the **Certificates** pane settings if you are using certificates for user authentication.

Note: Before you configure these settings, you should *install* (page [12](#)) and *configure* (page [67](#)) Reflection PKI Services Manager.

PKI server	<p>Specifies the name of the host running PKI Services Manager.</p> <p>The default is 'localhost', which indicates that PKI Services Manager is running on the same host as Reflection for Secure IT.</p> <p>If you're running PKI Services Manager on a different host, specify that host name or IP address.</p>
Port	<p>Specify the listening port used by PKI Services Manager. The default is 18081.</p>
Verify Connection	<p>Verifies that PKI Services Manager is installed and running on the specified host and that Reflection for Secure IT can access this server using the specified port.</p>
Launch PKI Manager	<p>This option is available only if PKI Services Manager is installed on the same host as Reflection for Secure IT.</p>
Public key file	<p>Identifies the public key used by Reflection for Secure IT to confirm the identity of Reflection PKI Services Manager. When you install PKI Services Manager, the key is installed to the default location.</p> <p>If you're running PKI Services Manager on a different host, copy the public key to the Reflection for Secure IT host and use this setting to specify the key location.</p>
Key comment	<p>Displays comment text, which includes identifying information about the PKI Services Manager key, if available</p>

SHA1 fingerprint	Displays the SHA1 hash for the PKI Services Manager key, if available.
MD5 fingerprint	Displays the MD5 hash for the PKI Services Manager key, if available.

Start and Stop the PKI Services Manager Service on Windows

The PKI Services Manager service starts automatically when you restart Windows. A command line utility (**winpki** (page 172)) is installed, which you can use to start, stop, restart, and check the status of the service. You can also start and stop the service using the Windows Services console.

To start the service

- Restart Windows.
-or-
- From a DOS command window, enter the following command:

```
winpki start
```


-or-
- Open the Windows Services console (Control Panel >Administrative Tools > Services), select Attachmate Reflection PKI Services Manager and click Start.

To stop the service

- From a DOS command window, enter the following command:

```
winpki stop
```


-or-
- Open the Windows Services console (Control Panel >Administrative Tools > Services), select Attachmate Reflection PKI Services Manager and click Stop.

To check the service status

- Start the PKI Services Manager console and look for status information on the status line at the bottom of the console window.
- or-
- From a DOS command window, enter the following command:

```
winpki ping
```
- or-
- Open the Windows Services console (Control Panel >Administrative Tools > Services) and view the status of Attachmate Reflection PKI Services Manager.

RSA SecurID Authentication

RSA SecurID is a two-factor authentication solution from RSA Security, Inc that is based on hardware or software tokens. We recommend that you review the Authentication Manager documentation before using SecurID.

Reflection for Secure IT supports RSA SecurID authentication using the Secure Shell keyboard-interactive protocol.

Requirements

You must have a correctly configured RSA SecurID environment.

Required Item	Function
RSA Authentication Manager	Verifies authentication requests and centrally manages authentication policies.
RSA Authentication Agent	Intercepts authentication requests and directs them to the Authentication Manager for authentication.
<hr/> Note: The RSA Authentication Agent for Windows or the RSA Authentication Manager must be running on the same computer as the Reflection for Secure IT server. <hr/>	
Hardware Token	A hardware device, such as a key fob or smart card, that generates a one-time authentication code.

How it works

The Reflection for Secure IT server acts as a SecurID client in order to authenticate a user.

1. The Reflection for Secure IT server receives a keyboard-interactive authentication request from a client.
2. If SecurID authentication is enabled, the Reflection for Secure IT server passes the user name to the RSA SecurID Agent.
3. The RSA SecurID agent returns a text prompt, which is sent to the client.
4. The client user responds to the prompt.
5. The Reflection for Secure IT server forwards this response to the RSA SecurID Agent, which may return another prompt. This continues until the RSA SecurID Agent indicates that authentication is complete.
6. If the RSA SecurID Agent indicates that authentication is successful, the client connection is allowed and the Reflection for Secure IT server provides user access based on the current server configuration. If the RSA SecurID Agent indicates that authentication failed, the client connection is not allowed.

Note: Authentication fails if a user is able to authenticate to the RSA SecurID Authentication Manager server, but no account exists for that user on the local computer or in the Windows domain.

Configure SecurID Authentication

When SecurID is configured, Reflection for Secure IT transfers control of authentication to the RSA Authentication Manager.

To configure the Reflection for Secure IT server

1. Install the RSA Authentication Agent on the computer running the Reflection for Secure IT server.
2. From the server console, click **Configuration** > **Authentication** > **RSA SecurID**.
3. Click **Allow** or **Require**.

Note: The server uses the keyboard-interactive authentication method to support RSA SecurID authentication. This is true even if keyboard-interactive authentication is disabled on the **Password** pane.

To configure the client

- Enable keyboard-interactive authentication. (This is the default for all Reflection for Secure IT clients.)

RSA SecurID Pane

Getting there

- From the server console, click **Configuration** > **Authentication** > **RSA SecurID**.

RSA SecurID authentication

Allow/Require/Deny The default is **Deny**.

Note: You must install an RSA Authentication Agent on the server computer before you can select **Allow** or **Require**.

Retries

Number of authentication attempts The default is 3.

Delay between tries (seconds) The default is 2.

RADIUS Authentication

RADIUS is an authentication, authorization, and accounting service that authenticates users by integrating with password databases, such as the a UNIX password file, Active Directory, LDAP, and simple text files containing user/password pairs. Reflection for Secure IT supports RADIUS for authentication purposes only.

Requirements

One or more RADIUS authentication servers must be configured. To configure Reflection for Secure IT, you need the name of the RADIUS server, the port used for RADIUS communication (usually 1812 or 1645), and the shared secret used by the RADIUS server.

How it Works

The Reflection for Secure IT server acts as a RADIUS client in order to authenticate a user. Requests are sent in order to each RADIUS servers you have configured on the RADIUS pane.

1. The Reflection for Secure IT server receives a keyboard-interactive authentication request from a client.

2. If RADIUS authentication is enabled, the Reflection for Secure IT server attempts to authenticate the user by sending an ACCESS-REQUEST message with the User-Name and Password attribute/value pair to the first RADIUS server you have configured.
3. The Reflection for Secure IT server waits for an ACCESS-ACCEPT or ACCESS-REJECT message from the RADIUS authentication server.
4. If the Reflection for Secure IT server receives an ACCESS-ACCEPT message, the client connection is allowed and the Reflection for Secure IT server provides user access based on the current server configuration. If the server receives an ACCESS-REJECT message, or it fails to receive a response, the server attempts to authenticate to any additional RADIUS servers you have configured. If no ACCESS-ACCEPT message is received from any RADIUS server, RADIUS authentication fails and the Reflection for Secure IT server attempts any other allowed authentications.

Note: Authentication fails if a user is able to authenticate to the RADIUS authentication server, but no account exists for that user on the Reflection for Secure IT server computer or the domain that the server is joined to.

Configure RADIUS Authentication

When RADIUS is configured, Reflection for Secure IT transfers control of authentication to the RADIUS authentication server.

To configure the Reflection for Secure IT server

- 1 From the **Password** pane, enable **Password authentication using keyboard interactive**. This is the default.
- 2 From the RADIUS pane, enable **Use RADIUS authentication**.
- 3 Click **Add**.
- 4 Specify your RADIUS server, the port used for RADIUS on that server, and the shared secret required for RADIUS clients to authenticate to that server.
- 5 Click **OK**.
- 6 Save your settings (**File > Save Settings**).

To configure the client

- Enable keyboard-interactive authentication. (This is the default for all Reflection for Secure IT clients.)

RADIUS Pane

Getting there

- From the server console, click **Configuration** > **Authentication** > **Password** > **RADIUS**.

Use these settings to transfer control of authentication to a RADIUS authentication server.

Use RADIUS authentication

This option is not available if keyboard interactive authentication is disabled on the **Password** pane (either **Password authentication** is set to **Deny** or **Password authentication using keyboard interactive** is unchecked).

Attempt local password authentication if RADIUS fails.

Enable this setting to allow users to log in locally if RADIUS authentication fails.

Authentication Servers

You can add one or more RADIUS servers to this list. Reflection for Secure IT attempts to authenticate client users by contacting the listed servers in order until a response to the authentication request is received.

Inherit servers

This option is available only if you are creating or editing a *subconfiguration* (page [129](#)). When it is selected (the default) inherited servers are listed after any other servers you have configured.

For example, if you are creating a user subconfiguration, you see globally configured servers at the end of the list. New servers added for the user subconfiguration appear above the globally configured servers.

Note: You can't delete or edit individual inherited servers.

RADIUS Server Dialog Box

Getting there

- 1 From the server console, click **Configuration** > **Authentication** > **Password** > **RADIUS**.
- 2 Select **Use RADIUS authentication**.
- 3 Click **Add** or **Edit**.

The options are:

Server	The name or IP address of the RADIUS authentication server.
Port	The port used for RADIUS requests.
Secret	The shared secret required to authenticate to the RADIUS server.

Note: This secret is stored as plain text in the configuration file. Protection for this file is provided via Windows Access Control Lists.

GSSAPI (Kerberos) Authentication

Reflection for Secure IT supports client authentication using Kerberos V5, a common *GSSAPI* (page [202](#)) implementation. No password is required, nor is it necessary to distribute keys or certificates. Windows uses Kerberos for network authentication, and Reflection for Secure IT integrates with the Windows Kerberos implementation.

When this method is enabled, both the client and server can obtain user tickets automatically from the Windows credential cache, and use these tickets for authentication.

Note: When GSSAPI is enabled for client authentication, you can also configure the Secure Shell connection to use Kerberos for server authentication.

Configure Client Authentication using Windows Credentials

If the server host computer and client users are members of the same Windows domain, you can use GSSAPI to authenticate client users. With this configuration, the user authenticates using his or her Windows domain credentials, and therefore doesn't need to enter a password to connect to the server. If the domain accounts are configured to be trusted for delegation, the user can access other domain resources as well, such as printers and file servers.

Note: This procedure describes how to configure just client authentication using Windows credentials — server authentication still requires the server host key. To use GSSAPI and Windows credentials for mutual authentication, see *Configure GSSAPI Server and Client Authentication* (page 52).

To configure Windows domain accounts

- 1 Add the server computer and client computers to the Windows domain.
- 2 Launch the Active Directory Users and Computers console and add the client users to the domain.
- 3 (Optional) If you want to use delegation of authentication, configure user account to be trusted for delegation (**Account** > **Account options** > **User is Trusted for delegation**).
- 4 (Optional) If you want to use delegation of authentication, configure the server computer properties to trust this computer for delegation (**General** > **Trust computer for delegation**).

To configure the Reflection for Secure IT server

- 1 Start the server console, and then click **Configuration**.
- 2 Go to **Authentication** > **GSSAPI / Kerberos V5**, and then select **Allow** or **Require**.
- 3 Save your settings (**File** > **Save Settings**).

To configure the Reflection for Secure IT client

- 1 Start Reflection for Secure IT Windows Client.
- 2 Open the **Reflection Secure Shell Settings** dialog box (**Connection** > **Connection Setup** > **Security**).
- 3 From the **General** tab, under **Authentication**, select **GSSAPI/Kerberos**.
- 4 From the **GSSAPI** tab:
 - Select **SSPI** (the default).
 - (Optional) If you don't want the client to forward the Kerberos ticket to the server, clear **Delegate credentials**.
- 5 Click **OK**.

The **Reflection Secure Shell Settings** dialog box closes.

- 6 When you configure the user for your client connection, you may need to include both the domain and user name using the format `domain\user`. This is required if the server computer has a local account name that matches your domain account. For example, if the local computer has a "joe" account and you log on using a domain account for "joe", you need to connect from the client as:

```
mydomain\joe
```

Note: Depending on your operating system, you may need to modify your system security settings to allow access to a terminal shell to users who authenticate using domain credentials. For more information, see [Command Shell Access in Windows](#) (page [124](#)).

GSSAPI / Kerberos V5 Pane

Getting there

- From the server console, click **Configuration** > **Authentication** > **GSSAPI / Kerberos V5**.

The Secure Shell protocol supports Kerberos authentication via GSSAPI (Generic Security Services Application Programming Interface). Reflection for Secure IT supports Kerberos authentication when the KDC is a Windows domain controller. Both the client user and server host must be part of the same Windows domain.

Allow/Require/Deny **Deny** is the default.

CHAPTER 8

Cached Credentials

In this Chapter

Understanding How Credentials Affect User Access to Resources	82
Best Practices for Using Cached Credentials	83
Record and Use Cached Credentials	86
Credential Cache Pane	87
Domain Access Pane	90
Add/Edit Credential Dialog Box	90
Select Credential Dialog Box	92
Filters Dialog Box	93

You can use cached credentials for any or all of the following:

- **User authentication**

This option is relevant only if users log into the server without using their Windows credentials (for example using public key authentication). Cached credentials provide these users with access to remote resources using their own Windows credentials. Without using cached credentials, users who log in with public key authentication have access to folders on local drives, but don't have access to network resources. For more information, see *Record and Use Cached Credentials* (page [86](#)).

- **Mapped drives and SFTP accessible directories**

This option allows you to provide access to *mapped drives* (page [95](#)) and *file transfer directories* (page [101](#)) that wouldn't be available to a user based on that user's own Windows account privileges.

- **Domain access**

This option allows you to specify a Windows account that the server can use when it queries Windows Active Directory for user attributes and group membership. For more information, see **Domain Access Pane** (page [90](#)).

Understanding How Credentials Affect User Access to Resources

For both file transfer and terminal sessions, access to remote directories (any location specified using a UNC path) can be affected by the user authentication method and the credential used for accessing that drive. This is summarized in the table below.

Caution: Be careful when configuring access with any credential other than the client user's own credential. When you configure an alternate credential to provide access to any folder on a server, Windows will allow access to other folders on the same server that are accessible to the alternate credential. For more information about this risk and how to handle it securely, see *Best Practices for Using Cached Credentials* (page [83](#)).

Notes:

- User access to directories for file transfers (**sftp** connections) is configured from **SFTP Directories** (page [106](#)). (**SFTP Directories** settings also apply to **scp** connections made using *SCP2* (page [206](#)). Depending on your configuration these directories may also apply to *SCP1* (page [205](#)) connections.)
 - User access to remote directories for **ssh** terminal sessions is configured using **Mapped Drives** (page [96](#)).
 - Access described here for password authentication also applies to sessions configured to use GSSAPI authentication. Access describe here for public key authentication also applies to other authentication methods (certificate, SecurID, RADIUS) for which the user doesn't provide Windows credentials during login.
-

Authentication method	Credential	Mapped drive or directory access
Password (default)	[Client user] (default)	The user sees both local and remote drives and directories that are allowed to that user's Windows account.
Public key	[Client user] (default)	If no credential cache is configured (the default), the user sees only local directories. If a drive or virtual directory is mapped to a remote network location, the user won't see that path, even if it is allowed for the user's account. If the Credential Cache (page 87) is configured to record and use credentials, the users sees both local and remote paths that are allowed for the user's account.
Password and Public key	Specific cached credential, for example: mydomain\Joe	The user has access to a directory if Joe's account has access to this location.

Best Practices for Using Cached Credentials

If you use cached credentials for SFTP directories or mapped drives, the best way to control user access to network resources is by using the default [Client user] option and configuring network access using Windows user and group settings.

Caution: If you decide to grant access to client users by specifying an alternate credential, you should review the information presented here to understand how to create a configuration that provides users with access to the data for which they are authorized, but does not grant them access to data for which they are not authorized.

Concerns

The following issues can affect user access when you specify a credential other than [Client user].

- Because of how Windows domains handle authentication, if you specify an account that can access multiple locations on the same server, knowledgeable client users with permission to create a terminal session (the default) can access all of those locations.

Note: In a Windows domain, where multiple physical file servers are configured to be accessed through a single host name, authentication and authorization are the same as if you are using a single physical file server.

- When new client connections are established using the session reuse feature (a default for Reflection Windows Secure Shell clients and FTP client sessions configured using the GUI), rights established in the original connection are available for all subsequent connections. This means that rights established for an SFTP connection will also be available in a terminal session.
- Client users can only update their own passwords. If the password for a specified alternate credential expires, other users will lose access to locations for which this credential is required until the password is updated in the credential cache database by the administrator or owner.
- The Reflection for Secure IT server will use only one alternate credential during a session to create drives or virtual directories on any given server. If you configure additional drives or directories on the same server using different credentials, some of these locations will not be available to the client user.

Recommended practices

Review these guidelines to help ensure that you are providing access to authorized data only.

- Use only the default [Client user] credential and control control user access to network resources using Windows user and group settings. This option is recommended.

If you use an alternate credential, use any or all of the following to help ensure that client users can't access unauthorized data by using the privileges associated with another user's credentials.

- Use a dedicated file server to provide data access for client users. Use alternate credentials only to provide access to this server; for all other network resources, limit access to the [Client user] credential.
- If you are providing access to a specific folder on a server that is used for other purposes, use a credential that has access only to that specific folder.
- If users require only SFTP access, disable access to terminal sessions using **Allow terminal shell** (page [125](#)).
- Use the same credential to access all drives or directories on any given server.

Sample scenarios

The following two scenarios involve two users, Mary and Joe, in an organization that has two folders, `downloads` and `payroll`, on the same server (`acme.com`).

Mary's account does not have access to any folders on the `acme.com` server.

Joe's account (`acme\joe`) has access to two locations on the `acme.com` server:

```
\\acme.com\downloads
```

```
\\acme.com\payroll
```

The following scenario shows how an administrator configuring mapped drives might open up a potential leak of information stored in the `payroll` folder.

Drive	Network path	Credential
O:	\\acme.com\downloads	acme\joe
P:	\\acme.com\payroll	[Client user]

When Mary connects, Joe's credentials are used to provide access to the O: drive. Although the P: drive is not mapped, Mary is still able to access the `payroll` folder (and any other folders on `acme.com` to which Joe has rights). For example, Mary can manually map a drive to `\\acme.com\payroll` from her terminal session without having to authenticate because she is already using Joe's credential, which gives her access to this folder.

To prevent this, the administrator should move the `downloads` folder to a different server and/or change the credential used for drive O: to a user who only has access to the `downloads` folder.

The next scenario shows how an administrator configuring SFTP directories might open up a potential leak of information stored in the `payroll` folder.

Virtual directory	Network path	Credential
downloads	\\acme.com\downloads	acme\joe
payroll	\\acme.com\payroll	[Client user]

When Mary logs onto the server using an SFTP client, Joe's credentials are used to provide access to the `downloads` directory. Mary's SFTP client session won't show the `payroll` directory. However, she might use the connection reuse feature to open a terminal session that will use the credentials that were already established for the SFTP connection. From this terminal session, Mary can access content in the `payroll` folder by manually mapping a drive using Joe's privileges.

To prevent this, the administrator should move the `downloads` folder to a different server, change the credential used for the `downloads` virtual directory to a user who only has access to the `downloads` folder, and/or disable access to the terminal shell.

Record and Use Cached Credentials

Use automatic credential caching if client users authenticate without using their Windows credentials (for example, with public key, certificate, or SecurID authentication), and also need access to domain resources that require domain credentials. Users must log in using a password at least once. User passwords are cached in an encrypted cache. After a user's password is cached, the server can use the cached password to acquire credentials on behalf of the user. This enables users to access domain resources.

Cached passwords can provide access to:

- SFTP virtual directories for which the physical directory is a network resource. (These directories apply to all SFTP and *SCP2* (page 206) connections, and may apply to *SCP1* (page 205) connections, depending on your configuration.)
- Mapped drives configured for use with terminal sessions.

Before you begin

- Configure and test the client and server to connect using public key authentication (or any other authentication method that doesn't require Windows credentials). At this point, client users can make connections to the server, but will not have access to domain resources that require their Windows credentials.

Configure the server

- 1 Start the server console, and then click **Configuration**.
- 2 From the **Credential Cache** pane, select both of the following:
 - **Record credentials in the cache when users log in**
 - **Use credentials in the cache for authentication**

Note: To enable **Use credentials in the cache for authentication**, you must select **Record credentials in the cache when users log in**. This is by design, and enables the server to update cached passwords when a password change is required.

- 3 From the **Password** pane, confirm that password authentication is set to **Allow** and **Allow password change** is selected. (This is the default configuration.)

- 4 From the **Public Key** pane (or the pane you used to configure an alternate authentication option), confirm that **Allow** is selected.

Note: If **Require** is selected, client authentication fails when **Use credentials in the cache for authentication** is enabled and no password is cached, or a cached password is no longer valid. If you want to use password caching and also require public key (or any alternate) authentication method, you need to use **Allow** initially. After users have logged in with their passwords, you can reset the value to **Require**. However, you will need to return this value to **Allow** whenever new passwords need to be cached, either because new users are logging in or passwords have expired.

- 5 Save your settings (**File > Save Settings**).

Configure and connect from the client

- 1 Confirm that the client is configured to attempt public key authentication first (if client users authenticate with public keys or certificates), and also supports Keyboard Interactive and/or Password authentication. (This is the default for Reflection for Secure IT clients.)
- 2 Connect to the server.

The first time the client user connects to the server, the server displays a password prompt. When the user enters his or her Windows credentials the connection is made and these credentials are saved to the credential cache. The client user now has access to domain resources that require Windows credentials. The user can make subsequent connections without entering the Windows password. When the password expires, the user is prompted for a new password and the cache is updated.

Credential Cache Pane

Getting there

- From the server console, click **Configuration > Authentication > Credential Cache**.

You can use cached credentials to manage access to network resources. Credentials are stored in an encrypted file in the *Reflection for Secure IT data folder* (page [205](#)).

To add credentials to the cache you can:

- Configure the server to record Windows credentials when users log in.
- Manually add user credentials to the cache.

You can use cached credentials for any or all of the following:

- **User authentication**

This option is relevant only if users log into the server without using their Windows credentials (for example using public key authentication). Cached credentials provide these users with access to remote resources using their own Windows credentials. Without using cached credentials, users who log in with public key authentication have access to folders on local drives, but don't have access to network resources. For more information, see *Record and Use Cached Credentials* (page [86](#)).

- **Mapped drives and SFTP accessible directories**

This option allows you to provide access to *mapped drives* (page [95](#)) and *file transfer directories* (page [101](#)) that wouldn't be available to a user based on that user's own Windows account privileges.

- **Domain access**

This option allows you to specify a Windows account that the server can use when it queries Windows Active Directory for user attributes and group membership. For more information, see **Domain Access Pane** (page [90](#)).

The options are:

Record credentials in the cache when users log in

When this item is selected:

- If a user authenticates using a Windows password, this credential is added to the cache.
- If a user is configured to authenticate using public key authentication (or any other method that doesn't require entering Windows credentials) and there's no credential for that user in the cache, the server authenticates the user the first time by requesting a password and then adds this credential to the cache. On subsequent logins, the server authenticates the user with the public key (or other method).
- If a user uploads a public key to the server using the Reflection for Secure IT Windows client upload utility and is prompted for a password during the upload, the credential is added to the cache at that time.

Use credentials in the cache for authentication

When this item is selected, users who authenticate using public keys (or any other authentication method that doesn't require entering Windows credentials) have access to domain resources using their own cached credentials.

Note: To enable **Use credentials in the cache for authentication**, you must select **Record credentials in the cache when users log in**. This is by design, and enables the server to update cached passwords when a password change is required.

Cache contents

Filters	<p>Opens the Filters (page 93) dialog box, which you can use to configure which credentials are listed.</p> <p>You can use a filtered view to manage your stored credentials. For example, if you want to remove all credentials last used before a specified date, you can set that filter, then remove all items in the filtered list.</p>
Refresh	<p>Refresh the display to match the current contents of the cache. (The display is also updated automatically when you launch the console, open this pane, or make edits to the cache contents.)</p>
Current filter	<p>The default is All credentials. Click Filters to change this filter. You can filter the list based on allowed uses and/or the last used date.</p>
User	<p>Shows the user account name in domain\user format.</p>
Last used	<p>Shows the date this account was last used for user authentication.</p> <hr/> <p>Note: The Last Used date is updated when a user logs in when Use credentials in the cache for authentication is selected. It is not updated when a credential is used for mapped drives, SFTP directories, or domain access.</p> <hr/>
Allowed uses	<p>Authentication The credential can be used to provide the user with access to remote domain resources using his or her own access rights (Use credentials in the cache for authentication is selected).</p> <p>SFTP Directories/Drive mapping The credential can be used for <i>SFTP accessible directories</i> (page 101) and <i>mapped drives</i> (page 95).</p> <p>Domain access The credential can be used by the server when it makes queries to Active Directory.</p>

Domain Access Pane

Getting there

- From the server console, click **Configuration** > **Authentication** > **Domain Access**.

From the **Domain Access** pane, you specify a Windows domain account that can be used to query Windows Active Directory for user attributes and group membership. You may need to specify an account if you do any of the following:

- Enable public key, certificate, SecurID, or RADIUS authentication for domain users without using password caching.
- Control access to the server based on domain group membership.
- Configure group-specific authentication settings based on domain group membership.

The specified credential is stored in the Reflection for Secure IT credential cache.

Whether you need this setting depends on your Active Directory configuration. When no account is specified from this pane (the default), the server queries Active Directory using the Local System account. If the Local System account doesn't have permission to read user attributes in Active Directory, the server attempts to use an anonymous logon to acquire Active Directory information. Anonymous logon is disabled by default starting with Windows Server 2003, and enabling it is not recommended. Under these conditions, the server is unable to acquire Active Directory information prior to user authentication; before you can use the features described above, you must specify a user account.

The options are:

- | | |
|--------------------------|--|
| Current Setting | Shows the currently configured domain access credential.

If this box is clear, the server queries Active Directory using the account the service is running under; the default is Local System. |
| Select credential | Opens the Select Credential (page 92) dialog box, which you can use to select an existing user account from the credential cache, or a new user. |
| Clear | Clears the current setting. This restores the default behavior. |

Add/Edit Credential Dialog Box

Getting there

This dialog box is available from the **Configuration** tab using any of the following command sequences:

- **Authentication** > **Credential Cache** > **Add** (or **Edit**)
- or
- **Authentication** > **Domain Access** > **Select Credential** > **Add** (or **Edit**)
- or
- **SFTP Directories** > **Add** > **Use a specific credential** > **Select Credential** > **Add** (or **Edit**)
- or
- **Mapped Drives** > **Add** > **Use a specific credential** > **Select Credential** > **Add** (or **Edit**)

Use this dialog box to add or edit credentials in the secure credential cache.

Note: The same cache is used to store credentials for multiple features. Which **Allowed uses** you see selected depends on how you got to the dialog box.

The options are:

- | | |
|--------------------|---|
| Domain\user | Specify the user using this format:
<code>domain\user</code>
If you omit the domain, the user is assumed to be local. |
| Password | Enter the Windows password for this user |
| Test | Click to see if the credentials are valid. |
| Last used | (Read only) displays the last time this credential was used for user authentication. |

Note: The **Last Used** date is updated when a user logs in when **Use credentials in the cache for authentication** is selected. It is not updated when a credential is used for mapped drives, SFTP directories, or domain access.

Allowed uses

User authentication

This option is relevant only if users log into the server without using their Windows credentials (for example using public key authentication). Cached credentials provide these users with access to remote resources using their own Windows credentials. Without using cached credentials, users who log in with public key authentication have access to folders on local drives, but don't have access to network resources.

Mapped drives and SFTP accessible directories

This option allows you to provide access to *mapped drives* (page 95) and *file transfer directories* (page 101) that wouldn't be available to a user based on that user's own Windows account privileges.

Note: When you open this dialog box from either the **Mapped Drives** or **SFTP Directories** pane, this option is selected by default and not configurable.

Domain access

This option allows you to specify a Windows account that the server can use when it queries Windows Active Directory for user attributes and group membership.

Note: When you open this dialog box from the **Domain Access** pane, this option is selected by default and not configurable.

Select Credential Dialog Box

Getting there

This dialog box is available from the **Configuration** tab using any of the following command sequences:

- **Authentication > Domain Access > Select Credential**
- or
- **SFTP Directories > Add > Use a specific credential > Select Credential**
- or
- **Mapped Drives > Add > Use a specific credential > Select Credential**

This dialog box shows a list of accounts available in the credential cache whose allowed use includes the feature you are configuring. You can select an existing name, change the filter to view additional names in the cache, or add a new account name.

- The user you select must be joined to the same domain as the server or to a domain that is trusted by the server's domain.
- The same cache is used to store credentials for multiple features. You can also view and modify credentials from the [Credential Cache](#) (page [87](#)) pane.

Caution: Be careful when configuring access with any credential other than the client user's own credential. When you configure an alternate credential to provide access to any folder on a server, Windows will allow access to other folders on the same server that are accessible to the alternate credential. For more information about this risk and how to handle it securely, see *Best Practices for Using Cached Credentials* (page [83](#)).

The options are:

Filters Open the [Filters](#) (page [93](#)) dialog box, which you can use to filter the view based on date or allowed use.

Current filter Shows the current filter setting. The default depends on how you got to this dialog box.

Add Add a user to the cache.

Note: If you try to add a user who is already in the cache, you'll get an error message. This may happen even though the credential is not currently visible in the list of available users. By default, cached credentials are not listed if they are not currently allowed for the feature you are configuring. To list additional users, click [Filters](#) and modify the filter settings.

Edit Edit the properties of the selected user.

Remove Remove the selected user from the credential cache.

Filters Dialog Box

Getting there

This dialog box is available from the **Configuration** tab using any of the following command sequences:

- **Authentication > Credential Cache > Filters**
or
- **Authentication > Domain Access > Select Credential > Filters**
or
- **SFTP Directories > Add > Use a specific credential > Select Credential > Filters**
or
- **Mapped Drives > Add > Use a specific credential > Select Credential > Filters**

Use this dialog box to filter your current view of the secure credential cache.

Last used Display credentials last used for authentication before (or after) the date you specify.

Note: The **Last Used** date is updated when a user logs in when **Use credentials in the cache for authentication** is selected. It is not updated when a credential is used for mapped drives, SFTP directories, or domain access.

Allowed uses Display credentials based on allowed use.

Notes:

- The default filter view depends on how you got to this dialog box.
 - Use the **Add/Edit Credential** (page 90) dialog box to change the allowed use for a credential.
 - When all **Allowed uses** are selected, all credentials will be listed (even those for which no allowed use is configured).
-

CHAPTER 9

Mapped Drives

In this Chapter

Configure Mapped Drives for Terminal Sessions	95
Mapped Drives Pane	96
Mapped Drive Settings Dialog Box	97

Configure Mapped Drives for Terminal Sessions

Mapped drives apply to terminal sessions.

By default, when a client user starts a terminal session, the user has access to local folders that are allowed for that user account. If the user authenticates without entering a Windows password (for example using public key authentication), that user needs to enter his or her credentials in order to access remote network resources (for example using **net use**).

You can use mapped drives to:

- Provide easy access to network locations that are not on the server.
- Provide users who authenticate without entering their Windows credentials with access to network resources using their own credentials. By configuring mapped drives, you enable users who authenticate using public key authentication to access remote resources without having to enter their Windows password.
- Provide users with access to network resources based on the rights associated with an alternate user.

To customize folder access for terminal sessions using mapped drives

- 1 From the server console, click **Configuration** > **Mapped Drives**.
- 2 Click **Add**.
- 3 For **Drive**, select an available drive letter from the drop-down list.

Note: In some cases, drive letters that are available from this drop-down list may conflict with drives that are already mapped for a particular client user. When this occurs, the drive you map on the Reflection for Secure IT server won't be available to that user.

- 4 For **Network path**, enter a value in UNC format (for example, \\computername\path\folder).

Note: In most cases the network path identifies a location on a remote server. You can also specify a shared folder on the local drive. This might be useful if you are using mapped drives to grant access based on an alternate user credential.

- 5 (Optional) By default **Use the credential of the client user** is selected. With this default option, the drive you specify is available to the client user if he or she has access rights to that network location. To grant access rights based on the rights associated with an alternate user, select **Use a specific credential**. The user you select must be joined to the same domain as the server or to a domain that is trusted by the server's domain.

Caution: Be careful when configuring access with any credential other than the client user's own credential. When you configure an alternate credential to provide access to any folder on a server, Windows will allow access to other folders on the same server that are accessible to the alternate credential. For more information about this risk and how to handle it securely, see *Best Practices for Using Cached Credentials* (page [83](#)).

- 6 Click **OK**.
- 7 Save your settings (**File > Save Settings**).

Mapped Drives Pane

Getting there

- From the server console, click **Configuration** > **Mapped Drives**.

Mapped drives are available to client terminal sessions.

Note: Items on this pane can be configured globally or as part of a *subconfiguration* (page [129](#)).

The options are:

Inherit drives This option is available only if you are creating or editing a *subconfiguration* (page [129](#)). When **Inherit drives** is checked, the client user inherits mapped drive settings from any applicable configuration higher in the following order of inheritance:

global
client host
group
user

For example, if you enable **Inherit drives** for a user and disable it for a group to which that user belongs, the user inherits directories configured for the group, but does not inherit client host and global drives.

Note: Inherited global drives show up in the directory list as read-only entries. Applicable group drives may also be visible as read-only entries. Inherited drives from client host subconfigurations are applied when the user connects, and are not visible in this list.

Drive The drive letter for this mapped drive.

Network path The UNC path for this drive.

Credential The user whose rights determine what access is granted.

[Client user] indicates that the user has access to drives based on the access rights of his or her own Windows account. If any other credential is specified the user is granted the rights associated with the specified credential.

Mapped Drive Settings Dialog Box

Getting there

- 1 From the server console, click **Configuration** > **Mapped Drives**.
- 2 Click **Add**.

Drive	The drive letter that will be available to terminal session users.
<hr/>	
<p>Note: In some cases, drive letters that are available from this drop-down list may conflict with drives that are already mapped for a particular client user. When this occurs, the drive you map on the Reflection for Secure IT server won't be available to that user.</p>	
<hr/>	
Network path	<p>This must be a UNC path. For example:</p> <pre>\\computername\path\folder</pre>
Use the credential of the client user	When this is selected (the default), the drive you specify is available to the client user if he or she has access rights to that network location.
Use a specific credential	Select this option to grant the client user rights associated with an alternate user whose name and password are stored in the <i>credential cache</i> (page 81).
Select Credential	Opens the Select Credential dialog box.
<hr/>	
<p>Note: The same cache is used to store credentials for multiple features. By default, when you open Select Credential from Mapped Drive Settings, you'll see any available credentials that can be used to provide access to mapped drives. You can select an available name from the list, add a new name, or change the filter to view additional credentials.</p>	
<hr/>	

CHAPTER 10

Secure File Transfer

In this Chapter

File Transfer Overview	99
Specify the User Login Directory	100
Customize Directory Access for File Transfers	101
Control Upload and Download Access	103
Virtual Root Directory in Reflection for Secure IT	103
Smart Copy and Checkpoint Resume	105
SFTP Directories Pane	106
Accessible Directory Settings Dialog Box	109

File Transfer Overview

The Reflection for Secure IT server supports secure file transfer using any of the following options:

- SCP1
- SCP2
- SFTP

You can allow or deny client access to these protocols using the **Permissions** pane.

Note: SCP1 is the SCP protocol used by OpenSSH. This protocol does not use the SFTP subsystem; it executes an `scp` command through the secure channel. SCP2 uses the SFTP protocol.

Use the **SFTP Directories** pane to modify file transfer directory access for users.

- **Specify the user login directory**

The default login directory is the Windows *user profile directory* (page [207](#)).

- **Create virtual directory names**

Use virtual directories to customize the directory names visible to users.

- **Configure which directories are accessible to users**

By default users have access to files in their Windows profile directory. You can change this default and/or add access to additional directories. You can also grant access rights based on the rights associated with an alternate user.

- **Configure upload and download access**

Limit user file access to one or more of the following: browse, download, upload, delete, and rename.

Specify the User Login Directory

The user login directory determines which directory's contents a user sees after connecting to the server. The default login directory is the Windows *user profile directory* (page [207](#)).

To change the user login directory

- 1 Click the **Configuration > SFTP Directories**.
- 2 Select a directory in the **User login directory** list.
- 3 Save your settings (**File > Save Settings**).

Notes:

- The specified login directory affects all SFTP and *SCP2* (page [206](#)) file transfers.
 - By default, changes to **User login directory** do not affect *SCP1* (page [205](#)) transfers. To apply the customized login directory to scp connections from OpenSSH clients, go to the **Permissions** tab and select **Use SFTP accessible directory settings for SCP1**.
 - To add an item to the **User login directory** list, add it to the **SFTP accessible directories** list and confirm that **Allow** is selected.
 - To have users log into their virtual root directory, which contains all accessible directories configured and allowed under **SFTP accessible directories**, select "<virtual root directory>".
 - When you change the user login directory, if the currently configured user public key directory does not correspond to the new login location, a warning message appears. The reason for this is that automatic key upload programs — including the one provided with the Reflection client — will not work correctly if public keys are not stored in a subdirectory called `.ssh2` in the user's login directory. You can ignore this message if you do not use public key authentication, or if you configure user keys manually.
-

Customize Directory Access for File Transfers

Use the **SFTP Directories** pane to customize directory access for file transfer. By default, when a client user starts an SFTP session, the user has access to files and directories located within the configured **Login directory** (the *Windows profile folder* (page 207) by default). You can configure SFTP directories to:

- Provide users with access to additional local or network resources using their own credentials.
- Provide users with access to network resources based on the rights associated with an alternate user.

Notes:

- Customized directory settings affect all SFTP and *SCP2* (page 206) connections.
 - By default, customized directories do not affect *SCP1* (page 205) connections. This means that users executing **scp** transfers from older OpenSSH clients have access to all files and folders allowed to them by the operating system, regardless of the current **SFTP Directories** settings. To apply customized directory settings to *SCP1* transfers, go to the **Permissions** tab and select **Use SFTP accessible directory settings for SCP1**.
-

To customize directory access

- 1 Start the server console, and then click **Configuration**.
- 2 Click **Permissions** and clear **Allow SCP1** to disable access using this protocol.
- 3 Click **SFTP Directories**.
- 4 Click **Add**.

The **Accessible Directory Settings** dialog box opens.

- 5 Specify virtual and physical directory values:

For	Do This
Virtual directory	Enter the directory name that you want your users to see; for example, <code>Downloads</code> .
Physical directory	<p>Enter the actual directory path; for example, <code>C:\Users\Downloads</code></p> <p>Use UNC paths to specify directories on remote servers; mapped drives are not supported.</p> <p>The following options are available for specifying user directories:</p> <p><code>%D</code> The user's <i>User profile folder</i> (page 207).</p> <p><code>%H</code> The user's <i>Home folder</i> (page 206).</p> <p><code>%u</code> The user's login name.</p> <p><code>%U</code> The user's domain name and login in the format <code>domain.username</code>.</p>

Note: Do not use `%u` or `%U` to point to a location within a user's Windows profile folder. Neither of these options works correctly for this purpose. Use these options to create your own user-specific locations in some other location, for example on a shared network file server. For details, see *Pattern Strings in Directory Paths* (page [165](#)).

- 6 (Optional) Modify the options under **Accessible directory permissions**. You can use this feature to limit user file access to one or more of the following: browse, download, upload, delete, and rename.
- 7 (Optional) By default **Use the credential of the client user** is selected. With this default option, the drive you specify is available to the client user only if he or she has access rights to that network location. To grant access rights based on the rights associated with an alternate user, select **Use a specific credential**. (This option is available only if the Physical directory specifies a UNC path.) The user you select must be joined to the same domain as the server or to a domain that is trusted by the server's domain.

Caution: Be careful when configuring access with any credential other than the client user's own credential. When you configure an alternate credential to provide access to any folder on a server, Windows will allow access to other folders on the same server that are accessible to the alternate credential. For more information about this risk and how to handle it securely, see *Best Practices for Using Cached Credentials* (page [83](#)).

- 8 Click **OK**.
- 9 Save your settings (**File** > **Save Settings**).

Control Upload and Download Access

By default, users have full access to the directories configured on the SFTP Directories pane. You can use directory permissions to limit what kinds of actions users can perform.

To configure accessible directory permissions

Note: This change affects both SCP2 and SFTP transfers.

- 1 Start the server console, and then click **Configuration**.
- 2 Click **Permissions** and clear **Allow SCP1** to disable access using this protocol.
- 3 Click **SFTP Directories**.
- 4 Select a directory and click **Edit**.
- 5 Configure **Accessible directory permissions**. For example, to allow users to view and download files, but disallow any changes to the server files, leave **Browse** and **Download** checked, and clear **Upload**, **Delete**, and **Rename**.
- 6 Click **OK**.
- 7 Save your settings (**File** > **Save Settings**).

Caution: Client users may have a number of ways to access server files and directories. Factors to consider when configuring your server include session access, tunneling access, and file and directory permissions configured on the system.

Virtual Root Directory in Reflection for Secure IT

The virtual root directory is the top-level directory that the user can see and access, containing all of the files and/or directories available to that user.

Using the Default Virtual Root Directory

By default, a user who connects to the Reflection for Secure IT server using **sftp** or **scp** has access to a virtual root directory that contains all the accessible directories available for that user profile. With this default configuration, you can limit access to multiple root-level directories.

In the following example, two accessible directories are configured and the user login directory is set equal to one of these directories:

Virtual directory	Physical directory
blue	c:\colors\blue
gray	c:\gray

User login directory = c:\gray

With these settings, the user can navigate up from the login directory to the virtual root directory, and from there view a list of all accessible directories, as shown in the following example from an **sftp** command window session:

```
/gray>pwd
Remote working directory: /gray
/gray>dir
.
..
black.txt
white.txt
/gray>cd ..
/>pwd
Remote working directory: /
/>dir
blue
gray
/>
```

With this configuration, **sftp** commands that use absolute paths need to include the accessible directory name (even when only one accessible directory is configured); for example:

```
/>get /gray/black.txt
```

Configuring a Chrooted Environment

A chrooted environment is equivalent to what you can configure on UNIX systems using the **chroot** command. In a chrooted environment, users have access to only the chroot directory and its subdirectories. The user login directory is always set equal to the virtual root directory. Users cannot navigate to any other directories.

To configure a chrooted environment, you use a forward slash (/) to specify the virtual directory; for example:

Virtual directory	Physical directory
/	c:\gray

User login directory = c:\gray (set automatically)

With this configuration, user access is limited to the c:\gray directory. (You can configure other accessible directories, but you will not be able to enable access to these directories while the chroot directory is enabled.) The user login directory is set automatically to c:\gray; no other option is available.

With these settings, the user sees the contents of `c:\gray` when he or she first logs in. The user *cannot* navigate from there to any higher directory, as shown in the following example from an **sftp** command window session:

```

/>pwd
Remote working directory: /

/>dir
.
..
black.txt
white.txt
/>cd ..
/>pwd
Remote working directory: /

/>

```

With this configuration, **sftp** commands that use absolute paths should *not* include the accessible directory name; for example:

```

/>get /black.txt

```

Providing Access to All Local Drives

You can set the physical directory equal to "\$drive" (not case-sensitive) to provide access to all local drives.

Virtual directory	Physical directory
/	\$Drive

With this configuration, the user sees each available drive as directory in the virtual root directory. Directory names are automatically generated using drive letters (C:, D:, and so on). In the following sample **sftp** session, two drives are available:

```

/>pwd
Remote working directory: /

/>dir
C:
D:

/>cd C:

/C:>pwd
Remote working directory: /C:

/C:>

```

Smart Copy and Checkpoint Resume

Reflection for Secure IT supports the following features that help minimize the amount of time and resources spent repeating unnecessary transfer of data.

Note: The Secure Shell client must also support this feature.

Identical Files

If a client user initiates a transfer and an identically named file already exists on the server, the server computes a hash of the server copy of the file and sends this value to the client. The client computes a hash of the client copy of the file and compares that to the value from the server. If the two hashes are identical, this indicates that the files are identical, and no data transfer occurs.

Automatic Resume of Interrupted File Transfer

Reflection for Secure IT client and servers can resume an interrupted file transfer at the point at which the transfer was interrupted. For example, if a connection is dropped during a file upload, the client user can restart the transfer. The Reflection for Secure IT client determines the size of the file on the server, and requests a hash of that file from the server. The client computes the hash of the local file up to the length that the server already has. If the hashes are the same, the transfer resumes at that point in the file.

Note: Computing a hash to compare files does not produce useful data for ASCII transfers between systems with different line endings, so the hash comparison is skipped in this case and the complete file is always transferred.

SFTP Directories Pane

Getting there

- From the server console, click **Configuration** > **SFTP Directories**.

Use the **SFTP Directories** pane to customize directory access for file transfer. By default, when a client user starts an SFTP session, the user has access to files and directories located within the configured **Login directory** (the *Windows profile folder* (page 207) by default). You can configure SFTP directories to:

- Provide users with access to additional local or network resources using their own credentials.
- Provide users with access to network resources based on the rights associated with an alternate user.

Note: Items on this pane can be configured globally or as part of a *subconfiguration* (page 129).

SFTP accessible directories

Inherit directories

This option is available only if you are creating or editing a *subconfiguration* (page 129). When **Inherit directories** is checked, the client user inherits directory settings from any applicable configuration higher in the following order of inheritance:

global
client host
group
user

For example, if you enable **Inherit directories** for a user and disable it for a group to which that user belongs, the user inherits directories configured for the group, but does not inherit client host and global directories.

Note: Inherited global directories show up in the directory list as read-only entries. Applicable group directories may also be visible as read-only entries. Inherited client host directories are applied when the user connects, and are not visible in this list.

Allow all

Use **Allow all** to select or clear the allow box for all listed directories.

Note: This option is not inherited by user or group subconfigurations.

Allow

Determines whether a listed directory is accessible to users. This option is selected by default when you create a new list item. Clear to leave an item on the list without providing access to the specified directory.

Virtual directory	The directory name that users see and access.
Physical directory	The actual directory path on the Reflection for Secure IT server or in the Windows domain.
Credential	The user whose rights determine what access is granted. [Client user] indicates that the user has access to directories based on the access rights of his or her own Windows account. If any other credential is specified the user is granted the rights associated with the specified credential.

User login directory

User login directory specifies which directory a user sees after connecting to the server using SFTP or SCP2. The default is %D, which specifies the *Windows user profile folder* (page [207](#)).

The list of available directories consists of <virtual root directory> and all currently configured and allowed directories.

- The value <virtual root directory> sets the login directory to be a virtual directory that contains all user-accessible directories.
- If you have configured a chrooted environment (**Virtual directory** = /), the user login directory is set automatically and can't be edited.

For additional information on virtual root and chroot directories, see *Virtual Root Directories in Reflection for Secure IT* (page [103](#)).

Notes:

- The customized directory settings you configure from the **SFTP Directories** pane affect all SFTP and SCP2 (page [205](#)) connections.
 - By default, customized directories do not affect SCP1 (page [205](#)) connections. This means that users executing **scp** transfers from older OpenSSH clients have access to all files and folders allowed to them by the operating system, regardless of the current **SFTP Directories** settings. To apply customized directory settings to SCP1 transfers, go to the **Permissions** tab and select **Use SFTP accessible directory settings for SCP1**.
 - The directory settings you configure from the **SFTP Directories** pane do not affect which directories are accessible from a terminal session. To ensure that users cannot access files using a terminal session, clear **Allow terminal shell** from the **Permissions** pane.
 - You can disallow all SFTP and SCP2 access by clearing **Allow SFTP/SCP2** from the **Permissions** pane. The **Permissions** pane setting overrides all **SFTP Directories** pane settings.
-

Accessible Directory Settings Dialog Box

Getting there

- From the server console, click **Configuration** > **SFTP Directories** > **Add** or **Edit**.

The options are:

Virtual directory The directory name that is visible to client users. For example:

uploads

Note: You can enter a forward slash (/) to configure a chrooted environment. Only the specified physical directory is available to users, and the user login directory is automatically set equal to that directory. If you set **Physical directory** to "\$drive" (not case-sensitive) rather than specifying a directory, this option provides access to multiple local drives. For details, see *Virtual Root Directories* (page [103](#)).

Physical directory The actual directory path on the Reflection for Secure IT server or in the Windows domain. For example:

C:\shared\uploads

Note: Use UNC paths to specify directories on remote servers; mapped drives are not supported.

You can specify any physical directory, or use one of the following pattern strings to specify user-specific directories. For details see *Pattern Strings in Directory Paths* (page [165](#)).

%D The user's *User profile folder* (page [207](#)).

%H The user's *Home folder* (page [206](#)).

%u The user's login name.

%U The user's domain name and login in the format domain.username.

Note: Do not use %u or %U to point to a location within a user's Windows profile folder. Neither of these options works correctly for this purpose. Use these options to create your own user-specific locations in some other location, for example on a shared network file server. For details, see *Pattern Strings in Directory Paths* (page [165](#)).

Permissions

Browse	View file and directory lists.
Download	View file contents.
Upload	Modify files, create files, create directories, and modify file attributes.
Delete	Delete files and directories.
Rename	Rename files and directories.

Credential

Use the credential of the client user When this is selected (the default), the virtual directory is available to the client user if he or she has access rights to the specified physical directory.

Note: This option is available only if **Physical directory** specifies a UNC path.

Use a specific credential Select this option to grant the client user rights associated with an alternate user whose name and password are stored in the *credential cache* (page [81](#)).

Select Credential Opens the **Select Credential** dialog box.

Note: The same cache is used to store credentials for multiple features. By default, when you open **Select Credential** from **Accessible Directory Settings**, you'll see all currently available credentials that can be used to provide access to file transfer directories. You can select an available name from the list, add a new name, or change the filter to view additional credentials.

CHAPTER 11

Controlling Access

In this Chapter

Access Control Settings	111
Using Allow and Deny Rules for Access Control	112
Controlling Access from Client Computers	114
Controlling Access by Group	118
Controlling Access by User	122
Command Shell Access	124
Permissions Pane	125

Access Control Settings

The table below provides an overview of server settings you can use to control client access to the server.

By default, all client users with an account on the server host (or an account in a common domain) can connect to the server using password authentication, open a terminal session, and access all local files and directories allowed by their credentials.

To	From the Configuration tab, click
Specify which client host computers can connect to the server	Access Control > Client Host Access Control
Specify which user groups can connect to the server.	Access Control > Group Access Control
Specify which individual users can connect to the server.	Access Control > User Access Control
Deny all logins	Permissions
Deny access to terminal sessions (support file transfer only)	Permissions
Deny access to non-interactive users (as configured in the local computer Security Policy)	Permissions
Configure port forwarding permissions	Permissions

To	From the Configuration tab, click
Specify which file transfer protocols are supported (SCP1, SFTP/SCP2)	Permissions
Limit the number of connections a user can make to the server.	General
Customize access to file transfer directories	SFTP Directories
Grant access rights for file transfers based on the rights associated with an alternate user	SFTP Directories > Add > Use a specific credential
Specify whether customized access to file transfer directories applies to <i>SCP1</i> (page 205) transfers	Permissions
Configure mapped drives (provide access to remote network locations during client terminal sessions)	Mapped Drives
Grant access rights for terminal sessions based on rights associated with an alternate user	Mapped Drives > Add > Use a specific credential
Provide access to remote domain resources for users who authenticate with public keys	Authentication > Credential Cache
Block IP addresses after multiple failed authentications	Authentication

Using Allow and Deny Rules for Access Control

You can control access to the server based on the client user name, the user's group membership, or the computer from which the user connects. For each of these categories, you can allow or deny access, or use a combination of allow and deny. You can specify rules for specific users, groups, or hosts, or use regular expressions to match multiple users, groups, or hosts with a single entry. Name matching is not case-sensitive.

The server first checks to see if access is allowed from the client host computer. If the client host is allowed, the server then checks both user and group rules to see if the client user is allowed access. For both host-based and group/user-based access control, the server uses the following logic to determine whether to allow a connection.

1. Check to see if any "Deny" rules are configured. If a client matches any denied expression, the connection is refused (even if the client also matches an allowed expression).

2. If the client does not match a denied expression, check to see if any "Allow" rules are configured.
 - If no "Allow" rules are defined, the client can connect.
 - If one or more "Allow" rules are configured on any pane, the client can connect only if the client matches one of the allowed expressions.

Examples

For the examples below, users are attempting to connect to a server with the following access control configuration. (No client host access items are configured.)

Group access settings:

Group	Access
administrators	Allow
contractors	Deny

User access settings:

User	Access
Joe	Allow
Don	Allow
Fred	Deny

Sample access with the configuration above:

User	Group Membership	Access?	Notes
Joe	users	yes	Joe is an allowed user and does not match any denied condition.
Don	contractors	no	Don is an allowed user, but is a member of a denied group.
Fred	administrators	no	Fred is in an allowed group, but is a denied user.
Paul	users	no	Allowed items are configured, but Paul does not match any allowed condition.

Controlling Access from Client Computers

Use the **Client Host Access Control** pane to control which client computers have access to the server. These settings apply to all users of the client computer. You can use either domain names or IP addresses to specify hosts. The value you enter is interpreted as a *regular expression* (page [152](#)).

You can allow or deny access, or use a combination of allow and deny. For information about how the server handles allow and deny rules, see *Using Allow and Deny Rules for Access Control* (page [112](#)).

Notes:

- For access lists that use domain names, the server always tries to resolve the client domain name. However, if name resolution fails, the server allows or denies access based on the client IP address. This means that even if a client's domain name is on the deny list, that client can connect when DNS lookup fails, unless its IP address is also on the deny list. To prevent access from hosts whose domain name could not be resolved, you can enable, from the **Network Binding** dialog box, **Require reverse DNS lookup**.
 - If IPv6 connections are supported, a client connecting using an IPv6 address may be allowed access even if the IPv4 address of that client is on the list of denied client hosts. To configure Reflection for Secure IT to deny all IPv6 (or IPv4) connections, from the **Network** (page [24](#)) pane, remove any listening address in IPv6 (or IPv4) format.
 - The resolved domain name for a client is the fully qualified domain name. This means that when you add a host to the allow or deny list using a domain name, you must either use a fully qualified domain name, or a regular expression, to ensure that host domain names are handled correctly. For example, if you deny access to the client "mypc", the client `mypc.myhost.com` will be able to connect. You must explicitly deny access to `"mypc\.myhost\.com"` or use an expression such as `"mypc\..*"` to ensure that this client is denied access.
-

Examples

In the following configuration, access to client hosts with an IPv4 address that begins with 123.156.78 is denied — users on any other client host (or users connecting from an IPv6 address) are allowed access.

Client host	Access
123\.156\.78\..*	Deny

In the following configuration, access to all hosts in the acme.com domain is allowed, *except* badpc — clients from any other domain are denied access.

Client host	Access
.*\acme.com	Allow
badpc.acme.com	Deny

The following configuration denies access to all hosts in the acme.com domain, *including* mypc — clients from any other domain are allowed access.

Note: Without the final line, no clients would be allowed access. This is because once any client is added to the Allow list, clients are allowed access only if they match an allowed expression.

Client host	Access
.*\acme.com	Deny
mypc.acme.com	Allow
.*	Allow

Client Host Access Control Pane

Getting there

- From the server console, click **Configuration > Access Control > Client Host Access Control**.

Use the **Client Host Access Control** pane to control which client hosts are allowed access to the server. Client host settings apply to all users of the client computer. You can allow or deny access, or use a combination of allow and deny. For information about how the server handles allow and deny rules, see *Using Allow and Deny Rules for Access Control* (page [112](#)). For additional information, including examples, see *Controlling Access from Client Computers* (page [114](#)).

Note: To change the sort order, click the column headings. Or, display the context menu (using a right mouse-click or SHIFT+F10) and click **Sort**.

Client host	A regular expression that specifies one or more domain names or IP addresses for client hosts. To add an item to the list, click Add .
<hr/> <hr/>	
	Note: Reflection for Secure IT always adds ^ to the beginning and \$ to the end of the regular expressions that you enter. This ensures that the regular expression matches the entire input.
Type	Indicates whether the host is specified using domain name or IP address.
Access	Indicates whether access is allowed or denied.

Client Host Access Control Dialog Box

Getting there

- From the server console, click **Configuration** > **Access Control** > **Client Host Access Control** > **Add**.

Use this dialog box to add client hosts to your allow or deny list. You can use either domain names or IP addresses to specify hosts. The value you enter is interpreted as a *regular expression* (page [152](#)).

Use a backslash before characters in the domain name that have a special meaning in regular expressions. For example, in regular expression syntax, a period acts as a wildcard character that matches any single character. To prevent periods in names and IP addresses from being interpreted as wildcards, precede them with a backslash (\). For example:

```
myhost\.mydomain\.com
```

Caution: Because a client host might be identified using a domain name, an IPv4 address, or an IPv6 address, you need to specify host names carefully. For additional information refer to the notes below.

The options are:

Fully qualified domain name	<p>Select to specify a host or hosts using the fully-qualified domain name. For example, to match all hosts at acme.com, select this option and enter:</p> <pre>.*\.acme\.com</pre>
Client IP address	<p>Select to specify a host or hosts using an IP address. The address can be in IPv4 or IPv6 format.</p> <p>Use \. to indicate a period in an IPv4 address to avoid unexpected wildcard matches. For example:</p> <pre>123\.45\.12\.45</pre> <p>If you specify an IPv6 address, don't use the condensed form of the address; the server matches the expression you enter here with the fully expanded IPv6 address (including all zeros). For example, if the client IPv6 address is:</p> <pre>ff06:0000:0000:0000:0000:0000:00c3</pre> <p>The following condensed address will <i>not</i> be a match.</p> <pre>ff06::c3</pre>
Allow connect	Add the host(s) to your list of allowed hosts.
Deny connect	Add the host(s) to your list of denied hosts.

Notes:

- The resolved domain name for a client is the fully qualified domain name. This means that when you add a host to the allow or deny list using a domain name, you must either use a fully qualified domain name, or a regular expression, to ensure that host domain names are handled correctly. For example, if you deny access to the client "mypc", the client mypc.myhost.com will be able to connect. You must explicitly deny access to "mypc\.myhost\.com" or use an expression such as "mypc\..*" to ensure that this client is denied access.
 - If IPv6 connections are supported, a client connecting using an IPv6 address may be allowed access even if the IPv4 address of that client is on the list of denied client hosts. To configure Reflection for Secure IT to deny all IPv6 (or IPv4) connections, from the **Network** (page 24) pane, remove any listening address in IPv6 (or IPv4) format.
 - Client domain names are not case sensitive (as specified in RFC 4343).
 - Reflection for Secure IT always adds ^ to the beginning and \$ to the end of the regular expressions that you enter. This ensures that the regular expression matches the entire input.
-

Controlling Access by Group

From the **Group Access Control** pane, you control which domain or local groups have access to the server. You can allow or deny access, or use a combination of allow and deny. For information about how the server handles allow and deny rules, see *Using Allow and Deny Rules for Access Control* (page [112](#)).

You can add groups to the list by specifying individual groups, or use regular expressions to match multiple groups. Group name matching is not case sensitive.

To ensure a greater degree of security, it is advisable to configure global settings that are more restrictive than group settings. With this model, you use group settings to increase, rather than decrease, access. Doing this helps to ensure settings that are more restrictive for a user whose group membership cannot be determined.

Notes:

- To specify a group that is a member of a Windows Active Directory domain, use either a single forward slash (/) or two backward slashes (\\) between the domain name and the group name.
- To specify a local group, either omit the domain name or use the local computer name as the domain name.
- To include a space in a group name, use [] (a space character enclosed in brackets). For example, to specify the Power Users group, use `Power[]Users`.

If you use a period (`Power.Users`), the expression matches the group name successfully. However, the expression also matches other group names that use any other character in place of the space (for example, `PowerXUsers`).

Examples

The following configuration denies access to any user who is a member of the local group called Red. Users from any other group are allowed to connect unless they match a deny list item on another pane.

Note: Users in the Red group are always denied access, even if they match an allowed item listed on this pane or on the [User Access Control](#) pane.

Group name(s)	Access
Red	Deny

The following configuration limits access to members of the Administrators group in the Acme domain. Other users are denied access unless they match an allowed user or client host rule.

Group name(s)	Access
Acme/Administrators	Allow

The following configuration allows access to all members of local and domain groups called Test and Developer.

Note: This configuration also allows access to groups that the administrator might not want to allow; for example, in addition to allowing access from Acme\Test, this configuration also allows access from Acme\NotTest and NotAcme\Test.

Group name(s)	Access
.*Test	Allow
.*Developer	Allow

By removing the wildcards, the following configuration ensures that access is provided only to the specific groups called Test and Developer on the local computer and in the Acme domain.

Group name(s)	Access
Test	Allow
Acme/Test	Allow
Developer	Allow
Acme/Developer	Allow

Group Access Control Pane

Getting there

- From the server console, click **Configuration** > **Access Control** > **Group Access Control**.

From the **Group Access Control** pane, you control which domain or local groups have access to the server. You can allow or deny access, or use a combination of allow and deny. For information about how the server handles allow and deny rules, see *Using Allow and Deny Rules for Access Control* (page [112](#)). For additional information, including examples, see *Controlling Access by Group* (page [118](#)).

Note: To change the sort order, click the column headings. Or, display the context menu (using a right mouse-click or SHIFT+F10) and click **Sort**.

Group

A regular expression that specifies one or more user groups. To add an item to the list, click **Add**.

Note: Reflection for Secure IT always adds ^ to the beginning and \$ to the end of the regular expressions that you enter. This ensures that the regular expression matches the entire input.

Access

Indicates whether access is allowed or denied.

Group Access Control Dialog Box

Getting there

- From the server console, click **Configuration > Access Control > Group Access Control > Add**.

From the **Group Access Control** dialog box, you can add rules to allow or deny group access. For additional information, including examples, see *Controlling Access by Group* (page [118](#)).

The options are:

Group name(s) A regular expression specifying one or more groups. Use a backslash before characters in the group name that have a special meaning in regular expressions.

To specify a Windows Active Directory domain, use either a single forward slash or two backward slashes between the domain name and the group name. The following formats are equivalent:

```
domain/group
domain\\group
```

Note: You must include the domain name for all domain groups. If you omit domain information, the server looks for groups defined on the local computer.

To specify a local group, use no domain name, or specify the local computer name for the domain name.

Allow Connect Allow access to the specified group(s).

Deny Connect Deny access to the specified group(s).

Notes:

- You can specify a comma-separated list of groups in the **Group name(s)** text box. When you click **OK**, each group in the list is added as a new line item in your list of configured groups.
 - Reflection for Secure IT always adds ^ to the beginning and \$ to the end of the regular expressions that you enter. This ensures that the regular expression matches the entire input.
-

Controlling Access by User

From the **User Access Control** pane, you control access to the server by individual users. You can allow or deny access, or use a combination of allow and deny. For information about how the server handles allow and deny rules, see *Using Allow and Deny Rules for Access Control* (page [112](#)).

You can add users to the list by specifying individual users, or use regular expressions to match multiple users.

Note: To specify a user that is a member of a Windows Active Directory domain, use either a single forward slash (/) or two backward slashes (\\) between the domain name and the user name. To specify a user defined on the local computer, omit the domain name, or use the local computer name as the domain name. To include a client host with the user name, use the format user@host.

Examples

The following configuration denies access to the user Joe, defined on the local computer, and the user Joe, in the Acme domain.

User name(s)	Access
Acme/Joe	Deny
Joe	Deny

The following configuration allows access to all members of the Acme domain, except Joe and Fred. Members of any other domain (including the local computer) are denied access.

User name(s)	Access
Acme/. *	Allow
Acme/Joe	Deny
Acme/Fred	Deny

The following configuration denies access to all users from the Suspect domain, *including* Fred. The user Fred matches two expressions, and the denied expression takes precedence. Users from any other domain (including the local computer) are allowed access.

Note: Without the final line, no users would be allowed access. This is because once any user is added to the Allow list, users are allowed access only if they match an allowed expression.

User name(s)	Access
Suspect/.*	Deny
Suspect/Fred	Allow
.*	Allow

User Access Control Pane

Getting there

- From the server console, click **Configuration** > **Access Control** > **User Access Control**.

From the **User Access Control** pane, you control access to the server by individual users. You can allow or deny access, or use a combination of allow and deny. For information about how the server handles allow and deny rules, see *Using Allow and Deny Rules for Access Control* (page [112](#)). For additional information, including examples, see *Controlling Access by User* (page [122](#)).

Note: To change the sort order, click the column headings. Or, display the context menu (using a right mouse-click or SHIFT+F10) and click **Sort**.

User	A regular expression that specifies one or more users. To add an item to the list, click Add .
Access	Indicates whether access is allowed or denied.

User Access Control Dialog Box

Getting there

- From the server console, click **Configuration** > **Access Control** > **User Access Control** > **Add**.

From this dialog box, you can add users to your allow or deny list. For additional information, including examples, see *Controlling Access by User* (page [122](#)).

The options are:

User name(s)

Enter a user name, or a regular expression that can match multiple users. You can specify just a user name, or include a client host or domain information. To include a client host with the user name, use the format `user@host`.

To specify a Windows Active Directory domain, use either a single forward slash or two backward slashes between the domain name and the user name. The following formats are equivalent:

```
domain/user
domain\user
```

When you omit domain information, the server looks for users defined on the local computer. You can also specify a local user by using the local computer name as the domain name.

Note: User Principal Name (UPN) format is not supported. If you specify "user@host", "host" is always the name of the client computer from which the user connects, not the name of the user's Windows domain.

Allow Connect

Allow access to the specified user(s).

Deny Connect

Deny access to the specified user(s).

Note: You can specify a comma-separated list of users in the **User name(s)** text box. When you click **OK**, each user in the list is added as a new line item in your list of configured users.

Command Shell Access

User access to the command shell (`cmd.exe`) can be controlled by both operating system settings and Reflection for Secure IT server settings. To configure server settings, use the **Permissions** (page [125](#)) pane.

Operating system security settings for command shell access vary with different Windows systems. For example, on many Windows 2003 Servers, default access to the command shell is restricted to administrators, members of TelnetClients group, and fully-authenticated users (that is, users who are logged on with a local password). With this configuration, users who *authenticate using domain credentials* (page [77](#)) won't have access to the command shell. Changes you can make to provide command shell access to these users include the following:

- Edit the security settings for `cmd.exe` to allow access to your users (or groups).
- Add your users (or groups) to the TelnetClients group.

Permissions Pane

Getting there

- From the server console, click **Configuration** > **Permissions**.

Notes:

- Changes you make on this pane do not affect permissions for existing client connections. You can restart the server to enforce these settings for all connections.
 - Items on this pane can be configured globally or as part of a *subconfiguration* (page [129](#)).
-

Caution: To ensure that the server launches the correct program for **Terminal provider** and **Exec request prefix**, use a fully-qualified path name and enclose any path name that includes spaces in double quotation marks. (If the executable or path name has a space in it, because of the way the Windows API function used by the server parses spaces, there is a risk that a different executable could be run. For details, see "Security Remarks" in the MSDN article at <http://msdn2.microsoft.com/en-us/library/ms682429>.)

Permission settings

- Deny all logins** Select to configure the server to deny all new client connections.
- This setting does not affect existing client sessions.
 - This setting is not available for subconfigurations. Use **Access Control** to control access by host, group, and/or user.

Allow terminal shell Specifies whether to allow client users access to a command window.

Note: You may also need to edit your operating system security settings to allow users access to a terminal shell. For more information, see *Command Shell Access* (page [124](#)).

Terminal provider Specifies which program to launch when a client connects to the server and **Allow terminal shell** is enabled. The program must be a text-based command-line utility. The default setting is `cmd.exe`, which launches a standard Windows DOS command window.

Terminal default directory Specifies the login directory for terminal shell sessions. You can specify any physical directory, or use one of the supported *pattern strings* (page [165](#)) to specify user-specific directories.

The default (%D) specifies the *user profile* (page [207](#)).

Allow exec requests Specifies whether to allow the client to execute commands on the server.

Exec request prefix This setting is available only when **Allow exec requests** is enabled. Use it to specify text to prepend to a command sent by the client.

Allow non-interactive users to log on Clear this setting to prevent non-interactive users from being able to connect to the server. Non-interactive users are those who do not have the right to "Allow log on locally" (or "Log on locally") as configured in the local computer Security Policy.

Note: On Windows 2003, non-interactive users will typically be unable to create a terminal shell even when they are allowed access. This is because of default restrictive permissions on the command prompt. This restriction is removed on Vista and Windows 2008.

File transfer

Allow SCP1

Clear to disable transfers using the SCP1 protocol. This protocol is used for **scp** commands from OpenSSH clients. The SCP1 protocol doesn't use the SFTP subsystem; it executes an **rcp** command through the secure channel.

Note: When **Allow exec requests** is enabled, SCP1 transfers are still possible, even if you have cleared this check box.

Use SFTP accessible directory settings for SCP1

Select to apply **SFTP Directories** (page [106](#)) pane settings to **scp** transfers from OpenSSH clients.

Allow SFTP/SCP2

Clear to disable transfers using SFTP and SCP2 (which use the SFTP subsystem).

Tunneling

Allow client to server (local) port forwarding

Clear to disable local port forwarding requests made by the client.

Allow server to client (remote) port forwarding

Clear to disable remote port forwarding requests made by the client.

CHAPTER 12

Working with Subconfigurations

In this Chapter

Subconfiguration Overview	129
Configure Settings Specific to a Client Host	130
Configure Group-Specific Settings	132
Configure User-Specific Settings	135
Revert settings to inherited values	138

Subconfiguration Overview

Using subconfigurations, you can apply settings to particular client hosts, users, and/or groups. Subconfiguration information is saved in the *server configuration file* (page [204](#)).

The server uses the following logic to apply subconfiguration settings.

1. Global settings apply to any client user who does not match a configured subconfiguration.
2. Host-specific settings override global settings.
3. Group-specific settings override host and global settings.
4. User-specific settings override host, group, and global settings.

When you configure subconfigurations, you'll see these items in the server panels:

Reload inherited settings	Removes subconfiguration-specific values from all settings on this pane. All settings values revert to their current inherited state.
----------------------------------	---

Note: This change is not finalized until you save your configuration using **File > Save**.

* (blue asterisk) Setting is no longer inherited	An asterisk indicates that the value of a setting is specific to the current subconfiguration. The server always applies the specified value, regardless of any subsequent changes you make to global or inherited settings.
---	--

Configure Settings Specific to a Client Host

Use client host subconfigurations to apply settings to all users connecting from a specified client computer.

To configure client host-specific settings

- 1 Start the server console, and then click **Configuration**.
- 2 Under **Subconfiguration**, click **Client Host Configuration**.
- 3 Click **Add**.
- 4 Identify the host computer using a fully qualified domain name or an IP address. *Regular expressions* (page [152](#)) are supported.

Caution: Have you correctly identified the client host computer or computers? Refer to the notes below.

- 5 Click any of the available panes, and then modify the settings you want to apply to users connecting from the specified computer or computers.

Note: When you change a setting in a subconfiguration panel, a blue asterisk appears next to that setting (*). This indicates that the setting is no longer inherited from the global configuration. If you change the value of a non-inherited setting to match the inherited value, it does not return the setting to the inherited state. Use **Reload inherited settings** to return pane settings to their inherited state.

- 6 Click **OK**.
- 7 Save your settings (**File** > **Save Settings**).

Notes

- The resolved domain name for a client is always the fully qualified domain name. This means that when you specify a host using a domain name, you must either use a fully qualified domain name, or a regular expression, to ensure that host domain names are handled correctly. For example, if you specify the client "mypc", settings will not apply to the client mypc.myhost.com. You must explicitly specify "mypc\myhost\.com" or use an expression such as "mypc\.*" to ensure that settings are applied to this host.
- If IPv6 connections are supported, a client connecting using an IPv6 address may be allowed access even if the IPv4 address of that client is on the list of denied client hosts. To configure Reflection for Secure IT to deny all IPv6 (or IPv4) connections, from the **Network** (page [24](#)) pane, remove any listening address in IPv6 (or IPv4) format.

- Always use `\.` to indicate a period to avoid unexpected wildcard matches. For example:
`myhost\.\mydomain\.\com`
`123\.45\.12\.45`

Client Host Configuration Pane

Getting there

- From the server console, click **Configuration** > **Subconfiguration** > **Client Host Configuration**.

From the **Client Host Configuration** pane, you can configure settings that apply to connections from specified client hosts. Subconfigurations in this category apply to all users who connect from any specified client computer.

Client host	A regular expression that specifies one or more domain names or IP addresses for client hosts. To add an item to the list, click Add .
Type	Indicates whether the host is specified using domain name or IP address.

Client Host Configuration Dialog Box

Getting there

- From the server console, click **Configuration** > **Subconfiguration** > **Client Host Configuration** > **Add**.

Use this dialog box to add client hosts to to apply settings to all users connecting from a specified client computer. You can use either domain names or IP addresses to specify hosts. The value you enter is interpreted as a *regular expression* (page [152](#)).

Use a backslash before characters in the domain name that have a special meaning in regular expressions. For example, in regular expression syntax, a period acts as a wildcard character that matches any single character. To prevent periods in names and IP addresses from being interpreted as wildcards, precede them with a backslash (`\`). For example:

```
myhost\.\mydomain\.\com
```

Caution: Because a client host might be identified using a domain name, an IPv4 address, or an IPv6 address, you need to specify host names carefully. For additional information refer to the notes below.

The options are:

Fully qualified domain name

Select to specify a host or hosts using the fully-qualified domain name. For example, to match all hosts at acme.com, select this option and enter:

```
.*\acme\.com
```

Client IP address

Select to specify a host or hosts using an IP address. The address can be in IPv4 or IPv6 format.

Use \. to indicate a period in an IPv4 address to avoid unexpected wildcard matches. For example:

```
123\.45\.12\.45
```

If you specify an IPv6 address, don't use the condensed form of the address; the server matches the expression you enter here with the fully expanded IPv6 address (including all zeros). For example, if the client IPv6 address is:

```
ff06:0000:0000:0000:0000:0000:00c3
```

The following condensed address will *not* be a match.

```
ff06::c3
```

Notes:

- The resolved domain name for a client is always the fully qualified domain name. This means that when you specify a host using a domain name, you must either use a fully qualified domain name, or a regular expression, to ensure that host domain names are handled correctly. For example, if you specify the client "mypc", settings will not apply to the client mypc.myhost.com. You must explicitly specify "mypc\.myhost\.com" or use an expression such as "mypc\.*" to ensure that settings are applied to this host.
 - If IPv6 connections are supported, a client connecting using an IPv6 address may be allowed access even if the IPv4 address of that client is on the list of denied client hosts. To configure Reflection for Secure IT to deny all IPv6 (or IPv4) connections, from the **Network** (page 24) pane, remove any listening address in IPv6 (or IPv4) format.
 - Client domain names are not case sensitive (as specified in RFC 4343).
-

Configure Group-Specific Settings

Use group subconfigurations to apply settings to all members of a specified group.

To configure group-specific settings

- 1 Start the server console, and then click **Configuration**.
- 2 Under **Subconfiguration**, click **Group Configuration**.
- 3 Click **Add**.
- 4 Specify the group type.

For a	Do this
Local group (configured on the server computer)	Select Local (the default).
Domain group (configured in Windows Active Directory)	Select Domain , and then specify the domain name.

- 5 In the **Group** box, type the name of the group.
- 6 Click any of the available panels and modify the settings you want to apply to this group.

Note: When you change a setting in a subconfiguration panel, a blue asterisk appears next to that setting (*). This indicates that the setting is no longer inherited from the global configuration. If you change the value of a non-inherited setting to match the inherited value, it does not return the setting to the inherited state. Use **Reload inherited settings** to return pane settings to their inherited state.

- 7 Click **OK**.
- 8 Save your settings (**File** > **Save Settings**).

Notes:

- Only one group subconfiguration can be applied to any given user. You can use the **Inheritance** options from the **User Configuration** dialog box to determine which group subconfiguration currently applies to a given user. You can control group precedence by changing the list order in the **Group Configuration** pane.
 - To ensure a greater degree of security, it is advisable to configure global settings that are more restrictive than group settings. With this model, you use group settings to increase, rather than decrease, access. Doing this helps to ensure settings that are more restrictive for a user whose group membership cannot be determined.
-

Group Configuration Pane

Getting there

- From the server console, click **Configuration** > **Subconfiguration** > **Group Configuration**.

From the **Group Configuration** pane, you can manage group-specific settings.

Note: Only one group configuration can be applied to any given user. If a user is a member of more than one group, the server uses the group highest on the list.

Group Type	Indicates whether the group is local, or if it belongs to a Windows Active Directory domain.
Group	The group name. To add an item to the list, click Add .
Domain	Indicates the domain name for groups that belong to a domain.
Move up Move down	Use these buttons to adjust the priority of a selected group.

Group Configuration Dialog Box

Getting there

- From the server console, click **Configuration** > **Subconfiguration** > **Group Configuration** > **Add**.

Use the following options to add a new group to the group subconfiguration list.

Group type	Choose from the following:	
	Select	If
	Local (the default)	The group is defined on the local computer.
	Domain	The group belongs to a Windows Active Directory domain.
Domain	Specify the domain name. This option is available only when Group type is set to Domain .	
Group	Specify the group name.	

Notes:

- Regular expressions are not supported for specifying domain and group information for subconfigurations.
 - Domain and group name matching is not case-sensitive.
 - After you have defined a group, click the available configuration panes (Permissions, Authentication, Password, Public Key, and SFTP Directories) to specify group-specific settings.
-

Configure User-Specific Settings

With user subconfigurations, you can to apply settings to individual users.

To configure user-specific settings

- 1 Start the server console, and then click **Configuration**.
- 2 Under **Subconfiguration**, click **User Configuration**.
- 3 Click **Add**.
- 4 Specify whether the user is defined on the local computer or as part of a domain:

To specify a	Do this
Local user (configured on the server computer)	Select Local (the default).
Domain user (configured in Windows Active Directory)	Select Domain , and then specify the domain name.

- 5 In the **User** box, type the user name.
- 6 (Optional) To see which group subconfiguration (if any) applies to the specified user, click **Inherited group**.
- 7 Click any of the available panes, and then modify the settings you want to apply to this user.

Note: When you change a setting in a subconfiguration panel, a blue asterisk appears next to that setting (*). This indicates that the setting is no longer inherited from the global configuration. If you change the value of a non-inherited setting to match the inherited value, it does not return the setting to the inherited state. Use **Reload inherited settings** to return pane settings to their inherited state.

- 8 Click **OK**.
- 9 Save your settings (**File** > **Save Settings**).

User Configuration Pane

Getting there

- From the server console, click **Configuration** > **Subconfiguration** > **User Configuration**.

From the **User Configuration** pane, you can manage user-specific settings.

User The user name. To add an item to the list, click **Add**.

Domain The user's domain.

Note: For users defined on the local computer, the domain is blank.

User Configuration Dialog Box

Getting there

- From the server console, click **Configuration** > **Subconfiguration** > **User Configuration** > **Add**.

Use the following options to add a new user to the user subconfiguration list.

Group type

Use the **Group type** options to add a new user to the user subconfiguration list.

The options are:

Local (the default) Select if this user is defined on the local computer.

Domain Select if this user is a member of a Windows Active Directory domain. Enter the domain name in the text box.

User Specify the user name.

Notes:

- Regular expressions are not supported for specifying domain and user information for subconfigurations.
- Domain and user name matching is not case-sensitive.
- After you have defined a group, click the available configuration panes (Permissions, Authentication, Password, Public Key, and SFTP Directories) to specify group-specific settings.

Inheritance

Only one group subconfiguration can be applied to any given user. Use the **Inheritance** options to determine which group subconfiguration (if any) currently applies to the specified user.

The options are:

Display group Click to update the **Inherited group** display based on the current server configuration and the current state of your local and domain groups.

Inherited group Displays which group configuration applies to the specified user. You must be logged in as a domain user to view this information.

Caution: The group membership shown here may be different from the group membership established when a user connects. When you are running the console, the server queries Active Directory using your current user account. During an actual connection, the server uses either the server account, or the domain user account specified from the **Domain Access** pane. If group membership cannot be determined at connection time, the server applies global settings.

Possible values are:

group name The group whose settings currently apply to this user.

Global No group subconfigurations currently apply to this user. Global settings will be applied.

Unknown Group membership could not be determined.

Notes:

Inherited group information is read-only. To modify group priorities, use the **Group Configuration** pane.

Because the server uses your current user account to access group information, you must be logged in as a user with access to the groups you have configured to view inherited group information; for example, you must be logged in as a domain user to view inherited information about domain groups.

Revert settings to inherited values

Notes:

- When you create a group or user subconfiguration, the value of any setting marked with an asterisk (*****) is no longer inherited. This means that the server always applies the specified value, regardless of any subsequent changes you make to global or inherited group settings.
 - If you change the value of a non-inherited setting to match the inherited value, it does not return the setting to the inherited state.
-

To revert settings to use inherited values

- 1 Open the **Group Configuration** (or **User Configuration**) pane.
- 2 Select the group (or user) you want to modify, and then click **Edit**.
- 3 Navigate to the pane whose settings you want to modify.

Note: Pay attention to the settings that are marked with an asterisk. The next step reloads inherited values for all of the settings on this pane. You will need to reconfigure any settings you want to keep in the subconfiguration.

- 4 Click **Reload inherited settings**, and then click **Yes** in response to the confirmation prompt.
- 5 Reconfigure any settings you want to keep in this subconfiguration.
- 6 Click **OK**.
- 7 Save your settings (**File > Save Settings**).

CHAPTER 13

Port Forwarding

In this Chapter

Port Forwarding Overview	139
Disable Port Forwarding	140

Port Forwarding Overview

Port forwarding, also known as tunneling, provides a way to redirect communications through the Secure Shell channel of an active session. When port forwarding is configured, all data sent to a specified port is redirected through the secure channel. Port forwarding is configured by the Secure Shell client, not the Reflection for Secure IT server. However, you can configure the server to enable or disable requests made by the client.

The client can request two kinds of port forwarding: local and remote.

Local Port Forwarding

In most cases, local port forwarding is used to forward data securely from another client application running on the same computer as the Secure Shell client. The Secure Shell client is configured to redirect data from a specified local port (on the same computer as the Secure Shell client), through the secure tunnel to a specified destination host and port. You can configure any other client running on the same computer to connect to the forwarded port (rather than directly to the destination host and port). After the Secure Shell connection is established, the Secure Shell client listens on the specified port and redirects all data sent to that port through the secure tunnel to the Secure Shell server. The server decrypts the data, and then directs it to the destination host and port.

Remote Port Forwarding

Remote port forwarding is used to forward data securely from any client application running on the same computer as the Secure Shell server. In this case, the client session requests that a specified remote port (on the same computer as the Secure Shell server) be used to redirect the data. You can configure any other client running on the same computer as the Secure Shell server to connect to the forwarded port (rather than directly to the destination host and port). After the Secure Shell connection is established, the Secure Shell server listens on the specified port and redirects all data sent to that port through the secure tunnel to the Secure Shell client. The client decrypts the data and then directs it to the destination host and port.

Disable Port Forwarding

By default, the server supports both local and remote port forwarding requests made by the client.

To disable port forwarding

- 1 Click the **Configuration** tab, and go to **Permissions**.
- 2 Under **Tunneling**, clear the boxes to disable local and/or remote forwarding.

CHAPTER 14

Troubleshooting

In this Chapter

Logging	141
Managing System Resources	146
Troubleshooting Group Settings	146
Troubleshooting Reflection for Secure IT Help	146

Logging

Reflection for Secure IT Windows Server supports two methods of logging: to the Windows Event Viewer and/or to a text log file.

- **Windows Event Viewer**

This method of logging is enabled by default. You can change the Windows Event Viewer logging settings from the [Event Logging](#) pane.

- **Log File**

To log to a text file, you must enable and configure the server from the [Debug Logging](#) pane.

You can configure the server to use either, both, or none of these logging methods. Both methods support the same options for configuring which events you want logged, and configuration for each method is independent of the other.

Use the Windows Event Viewer

By default, Reflection for Secure IT logs events to the Windows Event Viewer.

To view Reflection for Secure IT events from the Event Viewer

- 1 From the server console, go to **View** > [Event Viewer](#), or use the Event Viewer button:



- 2 Open the [Windows Logs](#) folder. (This step is not required on Windows XP and Windows Server 2003.)

- 3 Open the [Application](#) log and locate events from "Attachmate Reflection for Secure IT Server."

Note: You can archive content in Event Viewer logs, clear logs to record a particular series of events, and/or save log information to a variety of formats. For more information, see the Event Viewer Help.

Event Logging Pane

Getting there

- From the server console, click [Configuration](#) > [Event Logging](#).

Use [Event Logging](#) to configure logging to the Windows Event Viewer.

Notes:

- To open the Windows Event Viewer quickly from the server console use the Event Viewer button: 
-

The options are:

Enable logging to Windows Event Viewer

Enables logging to the Event Viewer. Use the log level options to determine which events are logged.

SSH server and SFTP event log level

Errors, Warnings, Information, Protocol details, Hex-dump

Use this list to determine which events are recorded in the event log. These categories provide increasing detail as you move down the list, and selecting any item automatically selects all the previous items. For full control of which events are recorded, use the [Custom](#) option.

Errors are fatal program errors, **Warnings** are authentication failures. **Information** includes all successful connections, logins, logouts, and general information. **Protocol details** include all messages sent to and from the server. **Hex-dump** includes all actual data exchanged and may include private and sensitive information. To maintain security, you should, at least, monitor **Errors** and **Warnings**.

Note: If you have configured Reflection PKI Services Manager to send debug messages to the Reflection for Secure IT server (by enabling client debugging on the PKI Services Manager server), you need to set the log lever to [Protocol details](#) or higher to see these messages.

Custom

Select **Custom** for full control over which events are recorded; click **Custom events** to specify the particular events or groups of events you want logged.

Enable Logging to a Text File

Text logging is not enabled by default. Enable logging to a debug log file to have logging messages sent to a simple text file.

You can use both the Windows Event Viewer and the debug log file, and you can configure different levels of logging to each. For example, you might configure the server to send more detailed logging information to the debug log file.

To enable logging to a text file

- 1 From the **Configuration** tab, click **Debug Logging**
- 2 Click **Enable debug logging to log file**, and then select the events you want logged.
- 3 Save your settings (**File** > **Save Settings**).

By default, logs are saved to a subfolder called `Logs` in the server's *data folder* (page 205). Log file names are generated automatically, using the format `RSSH-YYYYMMDD-HHMMSSmmm.log`, where `YYYYMMDD` indicates the date, and `HHMMSSmmm` indicates the GMT time of log file creation.

Debug Logging Pane

Getting there

- From the server console, click **Configuration** > **Debug Logging**.

Use **Debug Logging** to configure logging to a text file. You can use this log instead of, or in addition to the Windows Event Viewer. If you use both the Windows Event Viewer and a debug log, you can configure them to record at different logging levels.

Notes:

- When set to the same log level, the debug log contains most of the same information as the Windows Event Viewer, however the Event Viewer includes some events that occur before logging to the debug log begins.
 - You can configure the debug file to roll over based on size or time or both. If you configure both, log rollover occurs whenever the first threshold is reached.
 - Restarting the server always starts a new log.
-

The options are:

Enable debug logging to log file

Enables logging to a log file. Use the log level options to determine which events are logged.

SSH server and SFTP event log level

Errors, Warnings, Information, Protocol details, Hex-dump

Use this list to determine what is recorded in the debug log. These categories provide increasing detail as you move down the list, and selecting any item automatically selects all the previous items. For full control of which events are recorded, use the **Custom** option.

Errors are fatal program errors, **Warnings** are authentication failures. **Information** includes all successful connections, logins, logouts, and general information. **Protocol details** include all messages sent to and from the server. **Hex-dump** includes all actual data exchanged and may include private and sensitive information. To maintain security, you should, at least, monitor **Errors** and **Warnings**.

Note: If you have configured Reflection PKI Services Manager to send debug messages to the Reflection for Secure IT server (by enabling client debugging on the PKI Services Manager server), you need to set the log lever to **Protocol details** or higher to see these messages.

Custom

Select **Custom** for full control over which events are recorded. Click **Custom events** to specify which specific events or groups of events you want logged.

Log file information

Log file directory

Specifies the log file folder. Log file names are generated automatically, using the format RSSHD-YYYYMMDD-HHMMSSmmm.log, where YYYYMMDD indicates the date, and HHMMSSmmm indicates the GMT time of log file creation.

Log file rollover (by size)

Specifies that the log file should be closed and a new log opened when the file reaches the size you specify for **File size (MB)**.

Log file rollover (by time)	Specifies that the log file should be closed and a new log opened at regular intervals.
Base time (UTC)	Sets a base time, specified in <i>UTC (Universal Time, Coordinated)</i> (page 206), to use for triggering creation of a new log file. New files are created at this time and at even intervals during the day based on value you specify for Interval (hours) .
Interval (hours)	Determines the number of hours to wait before creating a new file. The value must be a whole number factor of 24.

For example, to have the log turn over twice a day starting at 2:00 PM Pacific Standard time, you would set **Base time (UTC)** to 22 (14:00 PST = 22:00 UTC) and **Interval (hours)** to 12.

Timestamps for log file entries	Use this setting to specify how times are recorded in the log file. The options are UTC or Local .
--	--

Custom Log Events Dialog Box

Getting there

- 1 Click the **Configuration** tab.
- 2 Click either **Event Logging** or **Debug Logging**.
- 3 Select **Custom**.
- 4 Click **Custom events**.

From the **Custom Log Events** dialog box, you can specify which events, or groups of events, you want logged. You can configure custom logging to the Windows Event Viewer and/or a text log file. Configuration for each logging method is independent of the other.

To configure logging for an individual event

- Click the check box for that event.

To configure logging for groups of events

- Select any combination of **Errors**, **Warnings**, **Information**, **Protocol details**, or **Hex-dump**.
- or-
- Highlight multiple events and press the SPACEBAR.

Note: To order the list by event name, click **Event Name**.

Managing System Resources

On servers with multiple simultaneous users, the memory available for Windows desktop heaps can become exhausted, and processes that require these resources will not be able to start. These include non-interactive processes such as services, scheduled tasks, and scripts.

The server starts a child process for every sftp session, scp transfer, terminal session, and exec request. Every session with one or more active child process uses the non-interactive Windows resources known as desktop heaps. To avoid exhausting the memory available for desktop heaps, you can use **Maximum number of connections** (on the **General** pane) to limit the number of possible connections. It is also possible to increase your system's capacity for non-interactive desktop heaps by decreasing the size of each heap. For details, refer to the Microsoft knowledge base.

Troubleshooting Group Settings

Settings changes that apply to Windows groups may take quite some time to propagate through the system. If you test a client connection immediately after configuring group settings, you may find that the client connection does not honor the changes you just made. Try waiting 10-20 minutes and test again.

Troubleshooting Reflection for Secure IT Help

The Reflection for Secure IT Help system runs in your default browser and uses JavaScript to support active content. If your browser settings don't support scripting, you may encounter one or more of the following problems:

- A security warning displays when you launch the help.
- The table of contents books do not open when you click the plus sign (+).
- No **Search** field is displayed on the **Search** tab.
- Clicking **Show Contents / Index / Search** has no effect.

To enable active content in Firefox

- 1 From the **Tools** menu in Firefox, select **Options**.
- 2 Click the Content icon.
- 3 Select the **Enable Java** check box.

To enable active content in Internet Explorer

- 1 From the **Tools** menu in Internet Explorer, select **Internet Options**.
- 2 Click the **Security** tab and select the Internet zone icon.
- 3 Click **Custom Level**.
- 4 Under **Scripting** > **Active Scripting**, select **Enable**.

Notes:

- If you require browser settings that do not support JavaScript, you can access all help content in PDF format from the documentation pages on the Attachmate Web site. See <http://support.attachmate.com/manuals/>.
 - From Microsoft Internet Explorer, the locally installed Reflection for Secure IT help pages run in the Internet Zone. This is because these pages include a "Mark of the Web" (`<!-- saved from url=(0014)about:internet -->`). Microsoft recommends use of this mark in application Help provided as HTML pages run from the local computer. For more information, see the MSDN knowledge base article at: <http://msdn2.microsoft.com/en-us/library/ms537628.aspx> (<http://msdn2.microsoft.com/en-us/library/ms537628.aspx>).
-

SECTION

Appendix

In this section

Files Used by Reflection for Secure IT	150
Regular Expression Syntax	152
Table of Migrated Settings	154
Table of Migrated PKI Settings	162
Manual Host Key Migration	164
Pattern Strings in Directory Paths	165
Keyboard Access to Console Features	167
winpki and pkid Command Reference	168
pkid_config Configuration File Reference	172
pki_mapfile Map File Reference	178
Sample Mapping Rules	185
Sample Map File with RuleType Stanzas	187
PKI Services Manager Return Codes	188
rsshhd Command Line Utility	191
ssh-keygen Command Line Utility	193
ssh-certtool Command Reference	196

APPENDIX A

Files Used by Reflection for Secure IT

Reflection for Secure IT stores files in the following location:

The default data folder location is:

Windows Server 2008:

`\ProgramData\Attachmate\RSecureServer\`

Windows Server 2003:

`\Documents and Settings\all users\Application
Data\Attachmate\RSecureServer\`

Note: The files in the data folder (with the exception of the host public key) contain information that should remain secure. These files should not be readable by any one except SYSTEM and Administrators. These file permissions are set by default.

Filename	Description
<code>rsshhd_config.xml</code>	Server configuration file. This file is in XML format. <hr/> <p>Note: To minimize the chance of introducing errors, we recommend using the console whenever you want to modify your server settings.</p> <hr/>
<code>hostkey</code>	The default private key of the public/private key pair used to identify the server to clients.
<code>hostkey.pub</code>	The default public key of the public/private key pair used to authenticate the server to clients.
<code>RSITDatabase</code>	The credential cache. The file is encrypted using AES 256. In addition, passwords within the database are encrypted using the same algorithm with a different, system-specific key unique to the user. Moving this file to another system is not supported unless the system is identical (such as in a failover environment).
<code>RSITDatabase.sec</code>	This file contains the key required to decrypt the credential cache and is required to use the cache. If it is deleted, you will need to recreate your credential cache.
<code>migration</code>	This hidden file indicates that the server has migrated settings from a prior version. When this file is present, the server won't repeat an automated migration. This file has no effect on migrations done using the <code>rsshhd</code> command line <code>-m</code> option.

Log Files

By default, Reflection for Secure IT stores log files in a `Logs` subfolder in the data folder.

Log File	Description
<code>Migration.Log</code>	Information about settings migration from an F-Secure or Reflection 6.x <code>sshd2_config</code> file.
<code>Console_Validation.log</code>	Information about invalid settings values in the <code>rsshd_config.xml</code> configuration file. This file is created when you start the console.
<code>Server_Validation.log</code>	Information about invalid settings values in the <code>rsshd_config.xml</code> configuration file. This file is created when you start the server.
<code>RSSHD-yyyymmdd-...log</code>	Debug log file. (These files are not created by default. Enable text logging using the Debug Logging pane.)

User-Specific Files

User-specific files control access to the server by individual client users. Reflection for Secure IT looks for user-specific files in the Windows user profile folder. The user profile folder is configurable by the Windows system administrator. The default is:

Windows Server 2003:
`\Documents and Settings\username\`

Windows Server 2008:
`\Users\username\`

File or Directory	Description
<code><user profile>\.ssh2</code>	Default user key directory. Copy user public keys to this directory.
<code><user profile>\.ssh2\authorization</code>	Default user authorization file. Add a line for each key using the format "key" followed by the public key name. For example: <code>key mykey.pub</code>

APPENDIX B

Regular Expression Syntax

Use Basic Regular Expression (BRE) syntax for settings that take regular expressions. This syntax is summarized in the following table.

Character	Meaning
.	Matches any single character.
[]	Indicates a character class. Matches any character inside the brackets (for example, [abc] matches "a", "b", and "c").
^	<p>If this metacharacter occurs at the start of a character class, it negates the character class. A negated character class matches any character except those inside the brackets (for example, [^abc] matches all characters except "a", "b", and "c").</p> <p>If ^ is at the beginning of the regular expression, it matches the beginning of the input (for example, ^[abc] will only match input that begins with "a", "b", or "c").</p>
-	In a character class, indicates a range of characters (for example, [0-9] matches any of the digits "0" through "9").
?	Indicates that the preceding expression is optional: it matches once or not at all (for example, [0-9][0-9]? matches "2" and "12").
+	Indicates that the preceding expression matches one or more times (for example, [0-9]+ matches "1", "13", "456", and so on).
*	Indicates that the preceding expression matches zero or more times.
??, +?, *?	Non-greedy versions of ?, +, and *. These match as little as possible, unlike the greedy versions that match as much as possible (for example, given the input "<abc><def>", <.*?> matches "<abc>" while <.*> matches "<abc><def>").
()	Grouping operator. Example: (\d+,)*\d+ matches a list of numbers separated by commas (for example, "1" or "1,23,456").
\	<p>Escape character: interpret the next character literally (for example, [0-9]+ matches one or more digits, but [0-9]\+ matches a digit followed by a plus character). Also used for abbreviations (such as \a for any alphanumeric character; see the following table).</p> <p>If \ is followed by a number n, it matches the nth match group (starting from 0). Example: <{.*?}>.*?</\0> matches</p>

Character	Meaning
	"<head>Contents</head>".
\$	At the end of a regular expression, this character matches the end of the input (for example, [0-9]\$ matches a digit at the end of the input).
	Alternation operator: separates two expressions, exactly one of which matches (for example, T the matches "The" or "the").
!	Negation operator: the expression following ! does not match the input (for example, a!b matches "a" not followed by "b").

The following abbreviations are also supported.

Abbreviation	Matches
\a	Any alphanumeric character: ([a-zA-Z0-9])
\b	White space (blank): ([\t])
\c	Any alphabetic character: ([a-zA-Z])
\d	Any decimal digit: ([0-9])
\h	Any hexadecimal digit: ([0-9a-fA-F])
\n	Newline: (\r (\r?\n))
\q	A quoted string: (\"[^\\"]*" \'[^\']*\'')
\w	A simple word: ([a-zA-Z]+)
\z	An integer ([0-9]+)

APPENDIX C

Table of Migrated Settings

When you install Reflection for Secure IT on systems with a Reflection 6.x server or F-Secure server, supported settings are migrated to the newer XML configuration file format. This table provides a summary of which settings are supported and how settings are migrated to the newer XML format.

Note: Settings for configuring certificate authentication are migrated when you install Reflection PKI Services Manager. For details, see *Table of Migrated PKI Settings* (page [162](#)).

sshd2_config Keyword	rsshd_config.xml Setting
AddGroupToToken	Not supported
AllowedAuthentications	Authentication.<xxx>.<xxx> Values: allow = 2, require = 3, deny = 1 gssapi-with-mic > GSSAPI. AllowGSSAPIAuthentication publickey > PublicKey.AllowPublicKeyAuthentication keyboard-interactive > KeyboardInteractive. AllowKeyboardInteractiveAuthentication password > Password.AllowPasswordAuthentication
AllowedPasswordAuthentications	Authentication.Radius.UseRadius
AllowGroups	GroupAccessControl.GroupEntry.GroupName. AllowAccess sets AllowAccess to true
AllowTcpForwardingForGroups	Not supported
AllowTcpForwardingForUsers	Not supported
AllowUsers	UserAccessControl.UserEntry.UserName. AllowAccess Sets AllowAccess to true
AllowHosts	ClientHostAccessControl.ClientHostServer. ClientDomain.AllowAccess Sets AllowAccess to true
AllowTcpForwarding	Permission.PermitC2SPortForwarding Permission.PermitS2CPortForwarding

<u>sshd2_config Keyword</u>	<u>rsshd_config.xml Setting</u>
AuthFailureErrorMessages	Authentication.AuthFailureErrorMessages
AuthImmediateDisconnect	Authentication.AuthImmediateDisconnect
AuthInteractiveFailureTimeout	Authentication.Password.Password-AttemptDelay
AuthKbdInt.NumOptional	Not supported
AuthKbdInt.Optional	Authentication.RSASecurID.RSASecurIDAuth entication Set to '2' if 'securid' is present in the migrated setting
AuthKbdInt.Plugin	Not supported
AuthKbdInt.Required	Authentication.RSASecurID.RSASecurIDAuth entication Set to '3' if 'securid' present in the migrated setting
AuthKbdInt.Retries	Not supported
AuthorizationFile	Authentication.PublicKeys.Authorization-File
AuthPublicKey.MaxSize	Authentication.PublicKeys.PublicKey-MaxSize
AuthPublicKey.MinSize	Authentication.PublicKeys.PublicKey-MinSize
BadKeyName	Not supported
BannerMessageFile	General.BannerMessageFile
CachePasswords	Authentication.UsePasswordCache
Cert.RSA.Compat.HashScheme	Not supported
Ciphers	Encryption.Ciphers.<xxx> aes128-ctr > aes128-ctr aes128-cbc > aes128-cbc aes128 > aes128-cbc aes192-ctr > aes192-ctr aes192-cbc > aes192-cbc aes192 > aes192-cbc aes256-ctr > aes256-ctr aes256-cbc > aes256-cbc aes256 > aes256-cbc 3des-ctr > not supported 3des-cbc > des3-cbc 3des > des3-cbc blowfish-ctr > not supported blowfish-cbc > blowfish-cbc blowfish > blowfish-cbc twofish > not supported

<u>sshd2_config Keyword</u>	<u>rsshd_config.xml Setting</u>
	arcfour > arcfour-128, arcfour-256, arcfour cast128-ctr > not supported cast128-cbc > cast128-cbc cast128 > cast128-cbc des-cbc@ssh.com > not supported des > not supported rc2-cbc@ssh.com > not supported none > NoEncryption Any > aes128-cbc, aes192-cbc, aes256-cbc, des3-cbc, blowfish-cbc, cast128-cbc, aes128-ctr, aes192-ctr, aes256-ctr, NoEncryption AnyStd > aes128-cbc, aes192-cbc, aes256-cbc, des3-cbc, blowfish-cbc, aes128-ctr, aes192-ctr, aes256-ctr AnyCipher > aes128-cbc, aes192-cbc, aes256-cbc, des3-cbc, blowfish-cbc, cast128-cbc, aes128-ctr, aes192-ctr, aes256-ctr AnyStdCipher > aes128-cbc, aes192-cbc, aes256-cbc, des3-cbc, blowfish-cbc, cast128-cbc, aes128-ctr, aes192-ctr, aes256-ctr
	<hr/> Note: If only unsupported ciphers are set, migration of ciphers setting will fail. <hr/>
CRLFile	Not supported
DefaultDirectory	Permission.TerminalDefaultDirectory
DenyGroups	GroupAccessControl.GroupEntry.GroupName.AllowAccess Sets AllowAccess to false
DenyHosts	ClientHostAccessControl.ClientHostServer.ClientDomain.AllowAccess Sets AllowAccess to false
DenyTcpForwardingForGroups	Not supported
DenyTcpForwardingForUsers	Not supported
DenyUsers	UserAccessControl.UserEntry.UserName.AllowAccess Sets AllowAccess to false
DisableVersionFallback	SSH1 not supported by Reflection for Secure IT
DoubleBackspace	Not supported
EmulationType	Not supported

sshd2_config Keyword	rsshd_config.xml Setting
EmulationTypeForCommands	Not supported
EmulationTypeForForcedCommand	Not supported
EnableLegacySubauthentication	Not supported
EventLogFilter	EventLogging.EventLoggingLevel DebugLogging.DebugLoggingLevel error - 1 error,warning - 2 error,warning,info - 3
FipsMode	Encryption.FipsMode
ForwardACL	Not supported
GSSAPI.AllowedMethods	Not supported
GSSAPI.DelegateToken	Not supported
HostCertificateFile	Identity.HostCertificateFile
HostKeyFile	Identity.HostKeyFile
HostKeyEkInitString	Not supported
HostKeyEkProvider	Not supported
HostKeyEkTimeOut	Not supported
HostSpecificConfig	Not supported
IdleTimeOut	General.IdleTimeout
IsPasswordChangeAllowed	Authentication.Password.PermitPasswordChange
KeepAlive	Network.Binding.TCPKeepAlive
LDAPServers	Not supported
LocalPki	Not supported
ListenAddress	Network.Binding.ListenAddress (first binding)
LogCertificateSubject	Not supported
LoginGraceTime	Authentication.GraceLoginTimeout
LogPublicKeyFingerPrint	Not supported
MACs	Encryption.MACs.<xxx> hmac-sha1 > hmac-sha1 hmac-sha256 > hmac-sha256 hmac-sha512 > hmac-sha512 hmac-md5 > hmac-md5

<u>sshd2_config Keyword</u>	<u>rsshd_config.xml Setting</u>
	hmac-sha256 > Not supported
	hmac-ripemd160 > hmac-ripemd160
	none > NoProtection
	Any > hmac-sha1, hmac-sha256, hmac-sha512, hmac-md5, hmac-ripemd160, NoProtection
	AnyStd > hmac-sha1, hmac-sha256, hmac-sha512, hmac-md5, NoProtection
	AnyMac > hmac-sha1, > hmac-md5, hmac-ripemd160
	AnyStdMac > hmac-sha1, hmac-md5
MapFile	Not supported
MaxBroadcastsPerSecond	Not supported
MaxConnections	General.MaximumConnection
NoDelay	Not supported
OCSPResponder	Not supported
PasswdPath	Not supported
PasswordGuesses	Authentication.Password.MaximumPasswordAttempts
PermitEmptyPasswords	Authentication.Password.PermitEmptyPassword
PermitRootLogin	Not supported
PermitUserTerminal	Permission.PermitTerminalShell
Pki	Not supported
PkiDisableCrls	Not supported
PkiOcspMode	Not supported
Port	Network.Binding.Port
PrivateWindowStation	Not supported
ProtocolVersionString	Identity.ProtocolVersionString
PublicHostKeyfile	Public key is copied - no XML setting
QuietMode	Not supported
RadiusKey	Authentication.Radius.RadiusServer.ServerSecret
RadiusServer	Authentication.Radius.RadiusServer.ServerName

<u>sshd2_config Keyword</u>	<u>rsshd_config.xml Setting</u>
RandomSeedFile	Not supported
RekeyIntervalSeconds	Encryption.KeyExchange.RekeyIntervalSeconds
RemoteCommandPrefix	Permission.ExecutionRequestPrefix
RequiredAuthentications	<p>Values: allow = 2, require = 3, deny = 1</p> <p>gssapi-with-mic > GSSAPI.AllowGSSAPIAuthentication</p> <p>publickey > PublicKey.AllowPublicKeyAuthentication</p> <p>keyboard- > KeyboardInteractive.AllowKeyboardInteractiveAuthentication</p> <p>password > Password.AllowPasswordAuthentication</p>
RequireReverseMapping	Network.Binding.RequireDNSLookup
ResolveClientHostName	Not supported
RevocationCa	Not supported
SettableEnvironmentVars	Not supported
Sftp-AdminDirList	Not migrated
Sftp-AdminUsers	Not migrated
Sftp-DirList	<p>SFTPDirectories.AccessibleDirectories.AccessibleDirectory</p> <p>Note:If a “/” chroot is defined, then this accessible directory will be marked allowed and others will be marked not allowed. Also, ‘Allow all’ setting will be unchecked.If multiple “/” chroot is found, migration only migrate the first entry of “/”.</p> <p>If no “/” chroot is defined, all accessible directory(s)will be marked allowed. Also, ‘Allow all’ setting will be checked.</p> <p>If the first entry of “/” chroot contains “\$Drive”, migration will NOT migrate ANY accessible directory(s).</p> <p>If a non-chroot accessible directory contains “\$Drive”, migration will skip this directory.</p>

Sftp-Home	SFTPDirectories.UserLoginDirectory <p>If Sftp-Home is empty, the server uses the first entry on Sftp_DirList, provided it is not a chrooted entry (forward slash).</p> <p>Note: If a “/” chroot is defined, then the user login directory will be set to “/” value.If multiple “/” chroot is found, then the first entry of “/” wins.If Sftp-Home directory is not one of accessible directory(s) or a child of one of the accessible directory(s), then user login directory will be set to “<virtual root directory>”</p>
SftpLogCategory	EventLogging.EventLoggingLevel DebugLogging.DebugLoggingLevel error,warning,info - 3
	<hr/> <p>Note: All SFTP log categories are now part of overall event/debug logging. By default, Error Warning Information logging levels provide at least the same or more information.</p> <hr/>
	User Login/Logout > error,warning,info - 2
	Uploads > error,warning,info - 2
	Downloads > error,warning,info - 2
	Directory Listings > error,warning,info - 2
	Modifications > error,warning,info - 2
SocksServer	Not supported
Ssh1Compatibility	SSH1 not supported by Reflection for Secure IT
Sshd1ConfigFile	SSH1 not supported by Reflection for Secure IT
Sshd1Path	SSH1 not supported by Reflection for Secure IT
SubAuthId	Not supported
Subsystem	Not applicable
Subsystem-sftp	Not applicable
TerminalProvider	Permission.TerminalShell
TryReverseMapping	Not supported
UserConfigDirectory	Authentication.PublicKeys.UserKey-Directory

UserSFTPDDirectory	Pre 6.0 F-Secure keyword setting maps to SFTPDirectories.UserLoginDirectory Uses same logic as Sftp-Home
UserSpecificConfig	Not migrated
VerboseMode	Not supported

APPENDIX D

Table of Migrated PKI Settings

When you use first run Reflection PKI Services Manager on a system with a Reflection for Secure IT 6.x or F-Secure server, settings are migrated from your configuration file (`sshd2_config`) and map files to the `pki_config` and `pki_map` files used by Reflection PKI Services Manager.

The following table summarizes how these prior versions settings are migrated.

Prior version keyword	Migrated?	Equivalent PKI Services Manager keyword
HostCA	Yes	TrustAnchor
HostCANoCRLs	Yes	TrustAnchor RevocationCheckOrder = none
HostCertificateFile	No	--
DynamicMapFile	Yes	DynamicFile (This keyword is configured in <code>pki_mapfile</code> .)
ExternalMapper	Yes	Supported in map file rules by using the <code>Extern</code> option in the conditional expression.
ExternalMapperTimeout	Yes	ExternTimeout (This keyword is configured in <code>pki_mapfile</code> .)
LDAPServers	Yes	CertServers CRLServers (All servers are migrated to both keywords)
LocalPKI	Yes	LocalStore
OCSPResponder	Yes	OCSPResponders
PkiOcspMode	Yes	RevocationCheckOrder
RevocationChecks	Yes	RevocationCheckOrder
RevocationCA	Yes	OcspCertificate

Prior version keyword	Migrated?	Equivalent PKI Services Manager keyword
MapFile	Yes	MapFile
OcspMode	Yes	RevocationCheckOrder
PKI	Yes	TrustAnchor
PkiDisableCrls	Yes	RevocationCheckOrder =none
PkiIgnoreBasicConstraints	Yes	StrictMode
SocksServer	No	--

APPENDIX E

Manual Host Key Migration

When you install Reflection for Secure IT on a computer that is already running an older version of the Reflection for Secure IT server (6.1 or earlier) or an F-Secure server, the upgraded server automatically copies your previous host key to the correct host key location and, if necessary, converts the key format.

If you have installed to a 64-bit system, and your existing host key uses a format that requires conversion, the automatic host key conversion may fail.

Note: To test the format of a converted key, go to the **Identity** pane, and then under **Host key**, click **Browse**, and double-click the host key name. If the upgrade was not successful, you'll see a message saying, "The selected file is not a valid private key."

If the automatic key migration was unsuccessful, use the following procedure to convert and install your host key:

To migrate a host key manually from an earlier version

- 1 Open a DOS command window.
- 2 Navigate to your F-Secure installation folder. For a default installation, the command is:

```
cd C:\Program Files\F-Secure\ssh server
```

- 3 Enter the following command:

```
./ssh-keygen2 -P -1 hostkey
```

Host key files named `hostkey_ssh2` and `hostkey_ssh2.pub` are created.

- 4 Copy the updated key to the name and location specified for **Private key** on the **Identity** pane, or update the server to use a different name and location.

APPENDIX F

Pattern Strings in Directory Paths

Reflection for Secure IT supports the following pattern strings for specifying directory paths for user public key locations and SFTP file transfer directories.

This string	Evaluates to
%D	<p>The user's <i>User profile</i> (page 207).</p> <p>For example, if the server is running on Windows Server 2003 and the user name is "joe", %D\.ssh2 will typically be equivalent to:</p> <pre>C:\Documents and Settings\joe\.ssh2</pre>
%H	<p>The user's <i>Home folder</i> (page 206).</p> <p>By default this is the same as the User profile, but the Windows system administrator can specify a different location.</p>
%u	<p>The user's login name.</p> <p>For example, if the user name is "joe", ssh_users\%u is equivalent to:</p> <pre>C:\ssh_users\joe</pre> <hr/> <p>Notes:</p> <ul style="list-style-type: none">▪ With this option, the server is unable to distinguish between a local and domain user with the same user name — both are given access to the same directory.▪ Do not use %u to specify a location in the Windows profile folder. Depending on how users have logged into the server host previously, the user-specific subdirectory in the profile path may be just a user name, or may be both a user and domain in the format "user.domain". <hr/>
%U	<p>The user's domain name and login in the format "domain.username".</p> <p>For example, if "joe" logs in from the "sky" domain (sky\joe), ssh_users\%U\ is equivalent to:</p> <pre>C:\ssh_users\sky.joe</pre> <hr/> <p>Note: Do not use %U to specify a location in the Windows profile folder. The format for specifying user and domain in the Windows profile path is "user.domain", which is the reverse of the order specified by %U.</p> <hr/>

Notes

- The value %D is equivalent to the Windows environment variable USERPROFILE.
 - The value %H is based on the Home folder value stored in the user's Windows account. The two Windows environment variables HOMEDRIVE and HOMEPATH are also based on this value.
 - The value of %u is equivalent to the Windows environment variable USERNAME.
 - The value of %U is based on two Windows environment variables: USERDOMAIN and USERNAME.
-

APPENDIX G

Keyboard Access to Console Features

All server console features can be controlled using the keyboard. The following list provides shortcuts for features that cannot be accessed using the Tab or arrow keys and have no visible keyboard alternative.

Feature	Location	Keyboard shortcut
Sort on a column	Access control panes	Shift+F10, S Use arrow keys to select the sort column.
Reload inherited settings	Subconfiguration panes	Alt+I
Restore pane defaults	All panes	Alt+A, D
View the Attachmate Software License Terms	About Reflection for Secure IT Server dialog box	Alt+V

APPENDIX H

winpki and pkid Command Reference

Use **winpki** (on Windows) or **pkid** (on UNIX systems) to configure, start, and stop the PKI Services Manager service, and to check certificate validity and allowed identities.

Synopsis

Windows:

```
winpki [command [command args]] [options...]
```

UNIX:

```
pkid [command [command args]] [options...]
```

command = **start** | **stop** | **restart** | **reload** | **ping** | **validate** <cert>

options = **[-b path]** **[-c cert]** **[-d level]** **[-f file]** **[-h]** **[-i]** **[-k]**

[-m path] **[-p]** **[-o key=value]** **[-t host]** **[-u user]** **[-V]** **[-w]**

Commands

start

Starts the service.

stop

Stops the service.

restart

Stops and restarts the service.

reload

Reloads the configuration without stopping the service. Note: Some settings require a restart.

ping

Displays service status and the port used by the service.

validate *certificate*

Validates a certificate and optionally provides information about allowed identities. The service must be running. For example, to determine if `sample.crt` is valid (UNIX syntax):

```
pkid validate sample.crt
```

Use **-u**, **-t**, or **-w** after the certificate name to get information about allowed identities for the specified certificate. For example, to determine if the user `joe` can authenticate using `sample.cer` (Windows syntax):

```
winpki validate sample.cer -u joe
```

Options

-b *path* **--baseDir** *path*

Specifies the data directory used for PKI Services Manager configuration.

-c *cert* **--cert** *cert*

Validates the specified certificate. This option is available when the service is not running. Use the **validate** command to validate certificates when the service is running.

-d *level* **--debug** *level*

Specifies the amount of information sent to the log. Allowed values are: 'error', 'warn', 'info', 'debug', and 'trace'. The default is 'error'.

-f *file* **--config_file** *file*

Launches using a non-default configuration file.

-h **--help**

Displays a brief summary of command options.

-i **--init**

This option is rarely needed. It initializes PKI Services Manager, which creates a key pair for the server, and creates user data directories and files. Initialization happens automatically during installation on UNIX systems and on first run on Windows systems. Using this option has no effect if your system is already initialized. Note: You can create new keys by deleting the existing keys (`pki_key` and `pki_key.pub`), and then using this option. Existing configuration files are not affected.

-k **--check-config**

Checks for errors in your configuration and map files and then quits.

-m *path* **--migrate** *path*

Migrates certificate authentication settings from Reflection and F-Secure configuration files. If *path* specifies a directory, PKI Services Manager looks for server (`sshd2_config`) and client (`ssh2_config`) configuration files in that directory and migrates settings from those files. If *path* specifies a file, PKI Services Manager migrates the settings in the specified file. Full path information is required for both files and directories. Note: If the `pki_config` file in the destination folder already has a trust anchor configured, no migration occurs. This helps ensure that the migration won't overwrite modifications you have already configured.

Settings are migrated to the `pki_config` and `pki_map` files used by PKI Services Manager. If you use the `-b` switch, files with your migrated settings are created in the specified directory. If you omit this switch, the files are created in the default PKI Services Manager configuration directory.

A migration log is created in the `logs` directory located in the PKI Services Manager data directory. By default, this log records at a level of 'info' which shows if errors or warnings occurred. The level can be elevated using `-d`.

`-o key=value --option key=value`

Sets any option that can be configured using a configuration file keyword. Options configured this way override configuration file settings. For a list of keywords and their meanings, see `pki_config`. Syntax alternatives are shown below. Use quotation marks to contain expressions that include spaces.

```
-o key1=value
-o key1="sample value"
-o "key1 value"
-o key=value1,value2
-o key="value1, value2"
```

To configure multiple options, use multiple `-o` switches.

```
-o key1=value -o key2=value
```

`-p--showkey`

Displays the public fingerprint and shows the full path and key name.

`-t host --hostName host`

Use this option after the certificate name following a `validate` command. PKI Services Manager reads the map file(s) and reports whether the specified host is an allowed identity for the host certificate being validated.

`-u user --userID user`

Use this option after the certificate name following a `validate` command. PKI Services Manager reads the map file(s) and reports whether the specified user is an allowed identity for the user certificate being validated. If you include a server name (in the form `user@server`), PKI Services Manager reports on whether the user is allowed to authenticate to the specified server. If you specify only a user name, PKI Services Manager tests whether the user is allowed to authenticate with this certificate without checking for host-specific conditions.

-V --version

Displays the product name and version.

-w [*host*] **--whoAmI** [*host*]

Use this option after the certificate name following a **validate** command. PKI Services Manager reads the identity map file(s) and returns a list of all allowed identities for the certificate being authenticated. If you specify a server name after this option, the list is limited to allowed users for connections to that server. If no server name is specified, PKI Services Manager doesn't check for server-specific conditions.

APPENDIX I

pkid_config Configuration File Reference

The Reflection PKI Services Manager console saves settings to the configuration file. You can also view and edit this file manually. The default file location is:

- UNIX
`/opt/attachmate/pkid/config/pki_config`
- Windows XP, Windows Server 2003:
`\Documents and Settings\all users\Application Data\Attachmate\ReflectionPKI\config\pki_config`
- Windows 7, Windows Vista, Windows Server 2008:
`\ProgramData\Attachmate\ReflectionPKI\config\pki_config`

File Format

The configuration file consists of keywords followed by values. The value can be separated from the keyword by tabs, spaces, or spaces and one '='. Any line starting with a pound sign (#) is a comment. Any empty line is ignored. Some keywords can appear multiple times, and these settings are applied cumulatively. Changes to settings do not take effect until you reload the settings or restart the service. (If a restart is required, that information is given in the keyword description.)

The file includes a global section that contains settings that apply to all validation queries. You can also create stanzas that configure certificate-specific settings. The **TrustAnchor** keyword marks the beginning of each trust anchor stanza. Settings beneath the **TrustAnchor** keyword apply only to that trust anchor. The stanza ends at the next **TrustAnchor** keyword.

Some settings must be configured outside any trust anchor stanzas. These settings apply to all validation queries. Where a setting is supported both globally and within a stanza, the value within the trust anchor stanza overrides the global value.

Keywords

AllowClientStats

Specifies whether PKI Services Manager allows clients to request PKI Services Manager runtime statistics. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'yes'.

AllowMD2Certificates

Allow certificates signed using the MD2 RSA hash. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'no'. You need to restart the service if you modify this setting.

AllowMD5InFipsMode

Allow certificates signed using the MD5 hash, even when FIPS mode is enabled. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'yes'. You need to restart the service if you modify this setting.

AllowVers1

Specifies whether PKI Services Manager allows version 1 certificates for a trust anchor. Note: Intermediate certificates must be version 3 regardless of the value of this setting. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'no'.

AllowWhoAmI

Specifies whether PKI Services Manager allows a client to query for the mapped identity (using **-w** or **--whoAmI**) when using PKI Services Manager to validate certificates. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'yes'.

CertSearchOrder

A comma-separated list that specifies where PKI Services Manager searches for intermediate certificates required to validate a certificate. Listed locations are searched in order. The options are 'local','certserver', 'aia', and 'windows'. The default is 'local, certserver.' (Note: If you select 'windows', PKI Services Manager uses only those certificates that are installed for use by the local computer, not certificates installed for the current user. To view and manage the local computer certificates, use the Microsoft Management Console. Add the Certificates Snap-in and configure it to manage certificates for the computer account.) Configure this keyword once, outside any stanza.

CertServers

Specifies a server from which PKI Services Manager can retrieve intermediate certificates when 'certserver' is included in the **CertSearchOrder** list. You can specify either an HTTP or an LDAP server. (For example: `ldap://certserver:10389` or `http://certserver:1080`) This keyword can be configured multiple times outside any stanza. The values are cumulative.

CRLServers

Specifies a server from which PKI Services Manager can retrieve Certificate Revocation Lists (CRLs) when 'crlserver' is included in the **RevocationCheckOrder** list. You can specify either an HTTP or an LDAP server. (For example: `ldap://crlserver:10389` or `http://crlserver:1080`.) This keyword can be configured multiple times outside of any stanza and multiple times per stanza. The values are cumulative.

ClientDebugging

Specifies whether the application that is requesting certificate validation can request and receive debug messages from PKI Services Manager. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'no'. Note: To view these messages you also need to set a sufficiently detailed debug level in the calling application. For the Reflection for Secure IT Windows server, specify "Protocol details" or higher. For the Reflection for Secure IT UNIX clients and servers, specify debug level 3 or higher.

EnforceDODPKI

Determines whether PKI Manager enforces settings that meet US Department of Defense PKI requirements. The allowed values are 'yes' and 'no'. The default is 'no'. When this setting is 'yes', the service will not start unless the following conditions are met: **FipsMode** = yes; **AllowMD2Certificates** = no; **AllowMD5InFipsMode** = no; **AllowVers1** = no; **CertSearchOrder** does not include 'windows'; and **RevocationCheckOrder** has at least one option specified and does not include 'none'.

ExplicitPolicy

Determines whether PKI Services Manager enforces application policies. This keyword can be configured once outside of any stanza and once per stanza. The allowed values are 'yes' and 'no'. The default is 'no'. If the value is 'yes' you must specify one or more application policies to be enforced using the **PolicyOID** keyword. Each application policy is specified with a Policy Identifier (OID). (Note: Policies may also be required by the certificate being presented or by a certificate within the chain of trust.)

FipsMode

Enforces security protocols and algorithms that meet FIPS 140-2 standards. The allowed values are 'yes' and 'no'. The default is 'yes'. Configure this keyword once, outside any stanza. You need to restart the service if you modify this setting.

KeyFilePath

Specifies the path to the private key used to identify Reflection PKI Services Manager. When no path is specified, the path or file name is relative to the PKI Services Manager configuration directory. Configure this keyword once, outside any stanza. This setting is required. If **KeyFilePath** is not specified, or no key is present, the PKI Services Manager service will not start. The default is 'pki_key'. You need to restart the service if you modify this setting. PKI Services Manager creates a key pair when it initializes the settings, but you can also use a key pair created by **ssh-keygen** (or another tool). Only RSA keys are allowed.

ListenAddress

Specifies the port on which PKI Services Manager listens for validation requests. The syntax is `host:port`. You can specify the host name using either an IP address or a host name. IP addresses can be in either IPv4 or IPv6 format. IPv6 addresses must be enclosed in square brackets, for example `[::D155:AB63]:18081`. The default is `0.0.0.0:18081`, which configures the server to listen on port 18081 using any available network adapter. This setting is required. You need to restart the service if you modify this setting.

LocalStore

The local store is used to hold items that are required for certificate validation. Depending on your configuration, this may include trusted root certificates, intermediate certificates, and/or Certificate Revocation Lists (CRLs). You can specify directories or files. When a directory is specified, all files in the specified directory and any subdirectories are included in the store. Files must be binary or base 64 encoded X.509 certificates or CRLs. This keyword can be configured multiple times outside any stanza. The values are cumulative. This setting is required.

LogFacility

Specifies the output location for log messages. Allowed values are 'file' and 'none'. The default is 'file'. Log files are created daily and saved to a directory called `logs` located in the PKI Services Manager data directory. Configure this keyword once, outside any stanza. You need to restart the service if you modify this setting.

LogLevel

Specifies the amount of information sent to the log. Allowed values are: 'error', 'warn', 'info', 'debug', and 'trace'. The log can contain both auditing messages (labeled "[audit]"), and debug messages (labeled "[debug]"). Auditing messages provide information about both successful and unsuccessful validation attempts. Debug messages are designed to help in troubleshooting. The default log level is 'error'. At this level, auditing messages are sent to the log, but debug messages are sent only if a PKI Services Manager error occurs, generally because PKI Services Manager is not correctly configured. The other options include audit messages plus increasing levels of detail in the debug messages. Configure this keyword once, outside any stanza.

MapFile

Specifies the location of the PKI Services Manager map file. Use the map file to configure which users or computers are allowed to authenticate with a valid certificate. When no path is specified, the path or file name is relative to the PKI Services Manager configuration directory. This setting is required. This keyword can be configured once outside of any stanza and once per stanza.

MaxLogFiles

Specifies the maximum number of log files to create. A new log file is automatically created daily. When the maximum is reached, the oldest log is removed. The default is 10. Configure this keyword once, outside any stanza. You need to restart the service if you modify this setting.

NetworkTimeout

Specifies the timeout for any network download: LDAP, HTTP, or OCSP. Units are milliseconds. The default is 20000. Configure this keyword once, outside any stanza. Configure this keyword once, outside any stanza.

OCSPCertificate

Specifies a certificate that can be used to verify the signature of the OCSP response. This is needed only if the OCSP response does not include the signer's certificate. The value can be either a certificate file or the Subject value of the certificate (for example `OcspCertificate = "CN = Secure CA, O = Secure Corporation, C = US"`). If you use the Subject value, the certificate must be in the local store. This keyword can be configured multiple times outside of any stanza and multiple times per stanza. The values are cumulative.

OCSPResponders

Specifies the address of an OCSP responder to use for checking certificate revocation when 'ocsp' is included in the **RevocationCheckOrder** list. Use an HTTP address to identify the responder. (For example: `http://ocsp.myhost.com:1080`.) This keyword can be configured multiple times outside of any stanza and multiple times per stanza. The values are cumulative.

PolicyOID

Specifies an allowed Policy Identifier (OID) to use when application policies are in force, either because **ExplicitPolicy** is 'yes' or because policies are required by the certificate being presented or by a certificate within the chain of trust. When **ExplicitPolicy** is 'yes', the specified OID must match at least one of the OIDs in the final policy set of the certificate chain. The value 2.5.29.32.0 allows use of any Policy Identifier. (Note: The default value is 'no-policy'. When **ExplicitPolicy** is set to 'yes', you must change **PolicyOID** to indicate which policy or policies are allowed; if **ExplicitPolicy** is set to 'yes' and **PolicyOID** is set to 'no-policy', no certificate can pass validation.) This keyword can be configured multiple times both outside any stanza and within a stanza. Configured values are cumulative.

RevocationCheckOrder

A comma-separated list that specifies which sources are used to check for certificate revocation and the order in which these checks occur. The options are 'ocsp', 'cdp', 'crlserver', 'local', and 'none'. The default is 'local'. Note: If you specify just 'none', no revocation checking occurs. If you specify 'none' with other options, PKI Services Manager attempts to determine the revocation status using the specified options until it reaches 'none'. If the certificate revocation status is still unknown at this point, authentication is allowed. This keyword can be configured once outside of any stanza and once per stanza.

StrictMode

Specifies whether strict checking rules (as defined in RFC 3280) are used when validating certificates. Many certificates cannot pass strict checks. The allowed values are 'yes' and 'no'. The default is 'no'. This keyword can be configured once outside of any stanza and once per stanza.

TrustAnchor

Specifies a certificate to use as the final trust point in a certificate chain of trust that Reflection for Secure IT validates. This can be an intermediate CA certificate, a root CA certificate, or a self-signed certificate (which can only validate itself). It can not be a user certificate or host certificate.. The value can be either a certificate filename or the contents of the Subject field defined in the certificate (for example `TrustAnchor = "CN = Secure CA, O = Secure Corporation, C = US"`). If you specify a certificate filename and include full path information, the trust anchor is used regardless of how you configure the **CertSearchOrder** keyword. If you specify a certificate filename without including full path information, **CertSearchOrder** must include 'local'; and PKI Services Manager looks for the certificate in your local store. If you specify the contents of the certificate's Subject field, **CertSearchOrder** must include 'local' and/or 'windows'; and PKI Services Manager looks for the certificate in your local store and/or Windows certificate store. This setting is required. To configure multiple trust anchors, add additional **TrustAnchor** lines.

Note: On Windows systems, you can view the Subject value of certificates in your store using the PKI Services Manager console. On UNIX systems, you can use `ssh-certview(1)` to view this information.

Any keywords under a **TrustAnchor** setting create a stanza. The values you configure within a trust anchor stanza are specific to that trust anchor.

APPENDIX J

pkimapfile Map File Reference

Reflection PKI Services Manager binds certificates to one or more allowed identities using mapping rules. Typically, allowed identities are users or hosts. To authenticate a user correctly, you need to define a rule that links information in the validated certificate to an allowed user account. The mapper provides flexible options for mapping certificates to names. You can specify allowed names explicitly in your rules, or define rules that extract information, such as user or host name, from a certificate. By using these options, you can bind identities to certificates without having to create a separate rule for each certificate.

The default map filename and location is:

- UNIX
`/opt/attachmate/pkid/config/pkimapfile`
- Windows XP, Windows Server 2003:
`\Documents and Settings\all users\Application Data\Attachmate\ReflectionPKI\config\pkimapfile`
- Windows 7, Windows Vista, Windows Server 2008:
`\ProgramData\Attachmate\ReflectionPKI\config\pkimapfile`

Note: On Windows systems, you can modify the map file from the Reflection PKI Services Manager console using the **Identity Mapper** pane.

File Format

The map file consists of keyword settings and rules. Each rule is a single line and is independent of other rules. The format of a rule is:

```
{Allowed-Identity} [Conditional Expression]
```

After a certificate is determined to be valid, rules are processed in order (based on rule type then sequence). If the certificate meets the requirements defined in the conditional expression (or if the rule has no condition), the allowed identities specified in that rule are allowed to authenticate. No additional rules are applied after the first match.

Within the map file, you can use the **RuleType** keyword to apply different mapping criteria based on whether a user or host presents the certificate. Note: Rule type determines the order in which rules are processed. The order for processing user certificates is: user-address, user, none. The order for processing host certificates is: host, none. Within each rule type, rules are processed in order from top to bottom.

Allowed Identity Set

The allowed identity set is a required component of a rule. Allowed identities can be specified using a combination of constant values and values extracted from the certificate. The set of allowed identities can take multiple constant values, extracted values, or a combination of both.

Using constant values to define allowed identities

Constant values are literal strings. Use white space to delimit separate values. (If an allowed name includes spaces, enclose it in quotes.) For example, the following rule uses literal strings to allow root, joe, and fred smith to authenticate with any valid certificate:

```
{ root joe "fred smith" }
```

The format "*domain\user*" is required for Windows domain users, for example:

```
{ windomain\joe "windomain\fred smith" }
```

Note: After PKI Services Manager determines that a certificate meets the condition defined in a rule, rule processing stops. In both examples above, no conditions are defined. This means the rule will be applied to any valid certificate and no subsequent rules will be processed. To create a similar rule, you would need to include all allowed identities within the same rule.

Two asterisks used alone { ** } act as a wildcard for defining the allowed identity set. This option may be useful for testing, but should otherwise be used only with extreme caution. If you use this wildcard in a user rule, any user presenting a valid certificate is allowed to authenticate to any user account on the server. This creates a major security risk by allowing access to accounts with root, administrator, or power user privileges without requiring a password. If you use this wildcard in a host rule, any server with a valid certificate is accepted by the client. If you do choose to use the wildcard, consider limiting access using other options:

- Use the wildcard only with certificates signed by Certification Authorities that you control.
- Use the wildcard only in rules that have very restrictive conditions.
- Use the wildcard only in server-specific user rules (those whose **RuleType** is **user-address**).
- Limit user account access on the server side. For example, on a Secure Shell server, you might define sftp chroot jails and allow no command shell or remote command access.

Using values extracted from the certificate

Use extracted values to construct the allowed identity set based on the contents of the certificate presented for authentication. Extracted values must be preceded and followed by "%". For example, to allow authentication by the host specified in the Host portion of the UPN field:

```
{ %UPN.Host% }
```

You can also combine literal strings with extracted identities. (You can prepend a literal string to an extracted identity, and/or append a literal string, but you cannot combine more than one extracted value to form a single identity.) The following example adds a Windows domain name to an extracted user identity:

```
{ windomain\%UPN.User% }
```

Note: If the extracted identity evaluates to an empty result, the entire concatenated string is deemed to be empty and is not included in the set of allowed identities. If the entire set of allowed identities is empty, the rule is deemed to have failed and processing continues to the next rule.

Supported certificate fields are:

Subject

The Subject field defined in the certificate. The comparison is done following X.500 rules (not as a string comparison). For a successful match, the format must follow standards described in RFC 2253. To be compliant with this standard, Subject and Issuer fields start with the Common Name (for example, "CN = Secure CA, O = Secure Corporation, C = US"). On UNIX systems, you can use the **ssh-certview** utility to obtain the Subject value in this format. On Windows systems, copy the Subject contents from the Details tab of the certificate viewer, paste to an editor, and then replace new line characters with commas.

Subject.CN

The Common Name portion of the Subject field, if present.

Subject.Email

The email attribute part of the Subject, if present.

DNS

The DNS part of a SubjectAltName, if present.

UPN

The "otherName" representation of the SubjectAltName field, with the OID of 1.3.6.1.4.1.311.20.2.3 (UPN OID), if present.

UPN.User

The userID portion of the UPN field.

UPN.Host

The host portion of the UPN field.

Email

The representation of SubjectAltName as defined in RFC 822.

Email.User

The userID portion of Email.

Email.Host

The host portion of Email.

SerialAndIssuer

The certificate serial number (hex encoded) and value of the certificate's Issuer field in this format:

serial_number Issuer

Use white space to separate the serial number from the issuer. For example:

461D07A8 CN = Secure CA, O = Secure Corporation, C = US

Cert

This indicates the entire certificate. The Operation must be **Equals** and the argument must be a file path to a certificate. Note: The Mapper does not use the certificate store defined by Reflection PKI Services Manager.

subst

This option is available when the conditional expression within a rule uses either **Regex** or **Extern**.

With **Regex**, use **subst** in combination with any regular expression that has a capturing group, which has been identified using round brackets (). If the regular expression includes an exact match to a specified certificate field, the value of the first capturing group in the expression replaces %subst% in the allowed identity set.

With **Extern**, use **subst** as a placeholder for the value returned by the external application.

Conditional Expression

When a conditional expression follows the {Allowed-Identity}, the allowed identities can authenticate only if the conditional expression is true. The use of a conditional expression is optional, but in most cases is recommended. If no conditional expression is included, the allowed identities can authenticate with any valid certificate.

After a certificate is determined to be valid, rules are processed in order (based on rule type then sequence). If the certificate meets the requirements defined in the conditional expression (or if the rule has no condition), the allowed identities specified in that rule are allowed to authenticate. No additional rules are applied after the first match.

The syntax for a conditional expression is:

Field Operation Argument

For *Field*, specify any of these supported certificate fields (described above): Subject, Subject.CN, Subject.Email, DNS, UPN, UPN.User, UPN.Host, Email, Email.Host, SerialAndIssuer, Cert, or subst.

For *Argument*, specify a string value.

For *Operation*, use one of the following:

Equals

Checks for absolute equality between the *Field* value and the *Argument* string. For DNS, UPN and Email options, the comparison is case-insensitive.

Contains

Checks if the *Field* value is contained anywhere within the *Argument* string. For DNS, UPN and Email options, the comparison is case-insensitive.

Regex

Applies the *Argument* as a regular expression to the *Field*. If the regular expression includes an exact match to the *Field* contents, the condition is true. If the set of allowed identities contains the string **%subst%**, the first capturing group (if defined) of the Regex match is inserted.

Extern

Uses an external application to test the condition. Use *Argument* to point to the application. Use **%subst%** in the allowed identity set as a placeholder for the value returned by the external application. If the match within the external application is successful, it should exit with status 0; a non-zero return means an unsuccessful match.

Sample rules with conditional expressions:

```
{ %UPN.Email% } Subject.CN Equals acme.com
{ joep } Subject Contains "Joe Plumber"
```

Rule Type Stanzas

Rule types apply different mapping criteria based on whether the validated certificate is a user certificate or a host certificate. Use the **RuleType** keyword to create a new stanza for each supported type. A stanza ends at the next **RuleType** keyword or the end of the file. The format is:

```
RuleType type
```

Valid rule types are:

none

The rule applies to both hosts and user certificates.

host

The rule applies to host certificates only.

user

The rule applies to user certificates only.

user-address = *server*

The rule applies only to user certificates authenticating to the specified server. Note: When PKI Services Manager evaluates a user-address rule, it uses the server name (not the DNS host name) of the server the user is connecting to. The server sends its name to PKI Services Manager when it requests validation of a user certificate, and PKI Services Manager uses that name when applying the user-address rule. To determine the host name that is sent, you can enter the **hostname** command from a Windows DOS window or from a UNIX terminal session.

For example, to create rules that apply only to users connecting to the server acme:

```
RuleType user-address=acme
```

Note: Rule type determines the order in which rules are processed. The order for processing user certificates is: user-address, user, none. The order for processing host certificates is: host, none. Within each rule type, rules are processed in order from top to bottom.

Keywords

DynamicFile

Specifies whether PKI Services Manager reloads the map file every time it checks for allowed identities. The allowed values are 'yes' and 'no'. The default is 'no'.

ExternTimeout

Sets the timeout for rules that use the **Extern** option.

RuleType

Marks the beginning of a rule type stanza, which can be used to apply different mapping criteria based on whether a user or host presents the certificate. The allowed values are 'user', 'host', 'none', and 'user-address = *server*'. The default is 'none'.

APPENDIX K

Sample Mapping Rules

Rule	What happens
<pre>{ guest }</pre>	Because no condition is included, all valid certificates are mapped to the user "guest". This can serve as a default rule. A rule like this should go at the end of the rule list to ensure that all other rules are processed first.
<pre>{ fred.jones } UPN.user Equals "fred"</pre>	If the UPN representation of SubjectAltName is present, and the user part is equal to "fred", the set of allowed identities is fred.jones.
<pre>{ %UPN.user% } UPN.host Equals "acme.com"</pre>	If a certificate has a UPN representation of SubjectAltName, and the host name part is "acme.com", the user name part of the UPN is returned as the set of allowed identities.
<pre>{ guest %UPN.user% }</pre>	If the UPN is set, the user part is included in the set of allowed identities (along with "guest"). Otherwise the set of allowed identities is "guest". Because there is no condition, this rule applies to any valid certificate.
<pre>{ fred root } Subject.CN Contains "Fred Jones"</pre>	If the CN of the certificate contains "Fred Jones", the set of allowed identities has two values: "fred" and "root".
<pre>{ %subst% } Subject.CN Regex [a-zA-Z\.\.]*([0-9])</pre>	Sets the allowed identity equal to the first numerical string within the common name portion of the Subject field. For example, if the CN is "joe.smith.12345", the allowed identity is set to "12345".
<pre>{ elmer.foo.com } Subject.CN Contains "elmer"</pre>	Sets the allowed identity to the fully-qualified domain name "elmer.foo.com" from a certificate that contains the short name "elmer".

Rule	What happens
<code>{ bob } Cert Equals /temp/certs/bob_cert.crt</code>	Compares the incoming certificate to the one locally stored. If they are equal, the allowed identity set is "bob".
<code>{ %subst% } Subject.CN Extern /bin/myapp</code>	PKI Services Manager sends the Common Name portion of the Subject field to the application "/bin/myapp". If the exit code of the called application equals 0, the allowed identity is set equal to the returned result.
<code>{ %UPN.User% } UPN Extern /bin/ldap-app</code>	In this case, an exit-code of 0 from the external application serves as confirmation that the UPN is an authorized user.
<code>{ %Subject.CN% %DNS% }</code>	Sets the allowed identity set to include the contents of either the Subject.CN field or the DNS part of SubjectAltName.
<code>{ windomain\%UPN.User% }</code>	Allows users from the specified Windows domain name to authenticate if their user name matches the UPN user name.

APPENDIX L

Sample Map File with RuleType Stanzas

```
RuleType user

# the following rules are evaluated for user certificates only:

{ scott } CN Contains acme

{ joe } CN Equals acme

{ guest }

RuleType host

# The following rule is evaluated for host certificates only:

{ elmer.acme } CN Contains elmer

RuleType user-address=myserver

# The following rule is evaluated only when myserver
# requests validation of a user certificate:

{ good %subst% } Regex UPN "([A-Za-z0-9\.-])@[*.]"

RuleType none

# "none" is the default if no RuleType is specified.
# If no rule is successfully applied from "user" or "host",
# this rule is evaluated.

{ good } SerialandIssuer contains 123 CN=foo
```

APPENDIX M

PKI Services Manager Return Codes

Reflection PKI Services Manager returns the following codes to the application requesting validation services.

- Code 0 = No errors, successful validation.
- Codes 1-10 = Command-line errors, either with **winpki** or **pkid**.
- Codes 11-19 = Network or protocol errors.
- Codes 21-29 = Validation errors.
- Codes 31-39 = Mapper errors (certificate is valid but could not be mapped).
- Codes 41-49 = CRL or other revocation errors

Code	Meaning
0	No errors.
1	General error, unknown cause.
2	Syntax error with the command, improper arguments.
3	PKI Services Manager is already running.
4	Error in the configuration file.
5	Timeout occurred while executing the command.
6	Network error (for example, cannot connect to PKI Services Manager).
7	Access denied, user does not have permission to run the command.
8	System error . This is an internal error. Re-run with <code>-d</code> switch to see what happened.
9	Migration or initialization failed. See migration error log.
11	Unknown command was requested by the calling application.
12	An exception was thrown by PKI Services Manager. For more information, see the PKI Services Manager event log.
13	Syntax error with the command or packet sent to PKI Services Manager.
14	Command was ignored (not currently used, internal error).

Code	Meaning
0	No errors.
15	Processing error. The certificate sent to PKI Services Manager is not encoded correctly.
16	Command failed (commands are: stop, reload, reconfigure).
17	Signature mismatch. Sender did not sign with a matching key.
18	Format error. The ASN protocol was not properly formatted
19	PKI Services Manager is in FIPS mode and the certificate is not valid in that mode
21	Certificate is invalid (expired, not signed, bad key, etc.)
22	No path. The issuing certificate could not be located.
23	Certificate is revoked.
24	No trust anchor. The path did not terminate to a known trust anchor.
25	Other validation error. Policy or other constraints failed.
26	Path length to the end certificate exceeded the CA path length constraint.
27	Certificate policy is invalid or does not match assertions in effect.
28	Certificate's signature does not match.
29	Unknown critical extension was encountered in a certificate or CRL.
31	Identity requested did not match allowed identities.
32	No identities are allowed for this certificate (no maps exist that match).
33	Calling application did not send an identity for matching (client-side error).
34	Certificate is valid, but requested WhoAmI processing
41	Unknown CRL processing error
42	No base for a delta CRL.
43	CRL has expired.
44	Cannot verify signature or it is bad.
45	Unknown CRL extension that is marked critical.

Code	Meaning
0	No errors.
46	Mismatch of IDP field in CRL.
47	No CRL available.

APPENDIX N

rsshd Command Line Utility

rsshd - Secure Shell server.

Synopsis

```
rsshd -start [-d LogLevel] [-f XmlConfigFile] [-p Port]
```

```
rsshd [-c [PasswordCacheFile]] [-h | -?] [-m [TextConfigFile] [-stop] [-V] [-w  
[DomainAccessFile] [F-SecureHostKeyFile]]
```

Description

You can use the **rsshd** command line utility to start and stop the Reflection for Secure IT server and manage server settings. To launch the service from the command line, use the **-start**.

Note: Without any parameters, **rsshd** starts a process and sends all log information to the command window without starting the service. In this case, the **rsshd** process runs with your current privileges, and remote connections will generally fail (unless you are running with highly elevated privileges).

Modification options (**-c**, **-d**, **-f**, **-m**, **-p**) are applied only to a new instance of the service.

Options

-c [*PasswordCacheFile*] [*F-SecureHostKeyFile*]

Migrates a password cache created using F-Secure or Reflection for Secure IT 6.x. If no password cache file is specified, **rsshd** looks for `rsitdapc` in the F-Secure installation folder. If the F-Secure private host key file is not specified, **rsshd** looks for the host key file based on information in the configuration file in the F-Secure installation folder. With this option the migration is the only action; the service isn't started.

-d *LogLevel*

Use in combination with **-start** to enable logging to a text log and set the log level. Log level values can be 1-5. These values correspond to the following in the console: Errors (1), Warnings (2), Information (3), Protocol details (4), Hex-dump (5) in the console. The default is 3. This setting overrides the configuration file setting.

-f *XmlConfigFile*

Use in combination with **-start** to start the service using the settings in the specified configuration file rather than the default configuration file.

-h | -?

Displays a short summary of command options. The service isn't started.

-m [*TextConfigFile*]

Migrates prior version settings and host key. Settings are converted from a text-based configuration file (created using F-Secure or Reflection for Secure IT prior to version 7.0) to the current default configuration file. If no configuration file is specified, **rsshd** looks for `sshd2_config` in the F-Secure installation folder. With this option the migration is the only action; the service isn't started.

-p *Port*

Specifies the port on which the server listens. The default is 22, which is the standard port for Secure Shell connections. Use this switch in combination with **-start**.

-start

Starts the service.

-stop

Stops the service.

-V

Displays product name and version information and exits. If other options are specified on the command line, they are ignored.

-w [*DomainAccessFile*] [*F-SecureHostKeyFile*]

Migrates Domain Access credentials from F-Secure. If `DomainAccessFile` is not specified, **rsshd** looks for `rsitdaun` in the F-Secure installation folder. If `F-SecureHostKeyFile` file is not, specified, **rsshd** looks for the host key file based on the F-Secure configuration file from the F-Secure installation folder.

Return values

rsshd returns 0 (zero) if the command completes successfully. Any non-zero value indicates a failure.

APPENDIX O

ssh-keygen Command Line Utility

ssh-keygen - Creation, management, and conversion of keys used for client and server authentication.

Synopsis

```
ssh-keygen [-b bits] -t type [-N [passphrase]] [-C comment] [-f output_keyfile]
```

```
ssh-keygen -B [-f input_keyfile]
```

```
ssh-keygen -c [-P passphrase] [-C comment] [-f keyfile]
```

```
ssh-keygen -e [-f input_keyfile]
```

```
ssh-keygen -p [-P old_passphrase] [-N new_passphrase] [-f keyfile]
```

```
ssh-keygen -i [-f input_keyfile]
```

```
ssh-keygen -y [-f input_keyfile]
```

```
ssh-keygen -l [-f input_keyfile]
```

Description

You can use the **ssh-keygen** command line utility to create RSA and DSA keys for public key authentication, to edit properties of existing keys, and to convert file formats. When no options are specified, **ssh-keygen** generates a 2048-bit RSA key pair and queries you for a key name and a passphrase to protect the private key. Public keys are created using the same base name as the private key, with an added `.pub` extension. The key location is displayed when key generation is complete.

Options

-b *bits*

Specifies the key size. Up to a point, a larger key size improves security. Increasing key size slows down the initial connection, but has no effect on the speed of encryption or decryption of the data stream after a successful connection has been made. The length of key you should use depends on many factors, including: the key type, the lifetime of the key, the value of the data being protected, the resources available to a potential attacker, and the size of the symmetric key you use in conjunction with this asymmetric key. To ensure the best choice for your needs, we recommend that you contact your security officer. Key sizes are rounded up to the next value evenly divisible by 64 bits. The default for DSA keys is 1024 bits; for RSA it is 2048 bits.

-B

Shows the fingerprint of the specified key in SHA-1 Bubble Babble format. You can specify the key file using **-f**. If you don't specify a file, you are queried for a filename. You can specify the private or public key name, but in either case, the public key must be available.

-c

Requests a change of the comment in the private and public key files. This operation is only supported for RSA1 keys. The program will prompt for the file containing the private keys, for the passphrase if the key has one, and for the new comment.

-C *comment*

Specifies information for the comment field within the key file. Use quotation marks if the string includes spaces. If you do not specify a comment when you create a key, a default comment is created that includes the key type, creator, date, and time.

-e

Uses the specified OpenSSH public or private key to generate a public key in Reflection format. You can specify the key file using **-f**. If you don't specify a file, you are queried for a filename.

-f *filename*

Specifies the filename for the generated private key. (A public key is also created and is always given the same name as the private key plus a `.pub` file extension.) This option can also be used in combination with **-e**, **-i**, **-l**, **-p**, **-y**, and **-B** to specify the input filename.

-i

Uses the specified Reflection public or private key to generate a public or private key in OpenSSH format. You can specify the key file using **-f**. If you don't specify a file, you are queried for a filename.

-h

Displays a summary of command line options.

-l

Show fingerprint of specified public key file using the MD5 hash. You can specify the key file using **-f**. If you don't specify a file, you are queried for a filename. If you specify a private key, **ssh-keygen** tries to find the matching public key file and prints its fingerprint.

-N *passphrase*

Sets the passphrase. For example, to specify the passphrase for a new key:

```
ssh-keygen -N mypassphrase -f keyfile
```

To create a new key that is not passphrase protected:

```
ssh-keygen -N -f keyfile
```

You can also use **-N** in combination with **-p** and **-P** to change the passphrase of an existing key.

-p

Use this option to change the passphrase of an existing private key. If you use this option alone, the program prompts for the file containing the private key, for the old passphrase, and twice for the new passphrase. You can use it in combination with **-f**, **-P**, and **-N** to change the passphrase non-interactively. For example:

```
ssh-keygen -p -f keyfile -P oldpassphrase -N newpassphrase
```

-P *passphrase*

Provides the (old) passphrase.

-q

Silence **ssh-keygen**.

-t *type*

Specifies the algorithm used for key generation. The possible values are "rsa" or "dsa" for protocol version 2.

-y

Uses the specified private key to derive a new copy of the public key. You can specify the key file using **-f**. If you don't specify a file, you are queried for a filename.

Return values

ssh-keygen returns 0 (zero) if the command completes successfully. Any non-zero value indicates a failure.

APPENDIX P

ssh-certtool Command Reference

SYNOPSIS

```
ssh-certtool [-b key-length] [-d debug-level] [-h] [-n algorithm] [-o output-  
file] [-p private-key] [-P] [--passphrase passphrase] [-V] [-z option]  
pkcs10|pkcs12 [arguments]
```

```
ssh-certtool [options] pkcs10 subject [keyUsage] [extendedKeyUsage]
```

```
ssh-certtool [options] pkcs12 [file1] ... [fileN]
```

DESCRIPTION

You can use **ssh-certtool** to create a PKCS#10 certificate request or to create a PKCS#12 package containing a private key and one or more certificates.

Creating a PKCS#10 certificate request

The general syntax for creating a PKCS#10 file is:

```
ssh-certtool [options] pkcs10 subject [keyUsage] [extendedKeyUsage]
```

Note: **req** is supported as a synonym for **pkcs10**.

The value you specify as subject defines the certificate's Subject field. The subject name is required. Use the distinguished name syntax specified by RFC2253. Use commas to separate Subject elements (RDNs). RDNs can be specified using standard abbreviations (CN) or OIDs (2.5.4.3). Quotation marks are required if the subject name contains embedded white space. For example, "CN=Steve Kille,O=Isode Limited,C=GB".

The filename of the generated certificate request is based on the prefix specified by the **-o** option, with `.pkcs10` appended. The default filename of a generated private key, when **-o** is not specified, is `output.pkcs10`.

To create a request using an existing private key use **-p** to specify the key. To generate a new private key for the request, you must specify either key type (**-n**), key size (**-b**) or both. The filename of the generated private key is based on the prefix specified by the **-o** option, with `.ssh2` appended. The default filename of a generated private key, when **-o** is not specified, is `output.ssh2`. If a key with the same name already exists, you are prompted to overwrite it. If you elect not to overwrite it, **ssh-certtool** exits with a return code of zero.

You can use optional flags to set `keyUsage` and `extendedKeyUsage` fields. Use commas, spaces or tabs to separate items. All Key Usage and Extended Key Usage flags are marked as critical in the PKCS#10 request. Valid `keyUsage` flags are `digitalSignature`, `nonRepudiation`, `keyEncipherment`, `dataEncipherment`, `keyAgreement`, `keyCertSign`, `cRLSign`, `encipherOnly` and `decipherOnly`. If you omit this argument, the `digitalSignature` and `keyEncipherment` flags are set by default. Valid `extendedKeyUsage` flags are `anyExtendedKeyUsage`, `serverAuth`, `clientAuth`, `codeSigning` and `emailProtection`. No extended key usage flags are set by default.

Creating a PKCS#12 package

The general syntax for creating a PKCS#12 package is:

```
ssh-certtool [options] pkcs12 [file1] ... [fileN]
```

This constructs a PKCS#12 package file containing one private key and multiple certificates read from the *file* arguments. The PKCS#12 package file contains one safe, which contains the private key and all the certificates. The filename of the generated package file is based on the prefix specified by the `-o` option, with `.p12` appended. The default filename of the generated PKCS#12 package, when `-o` is not specified, is `output.p12`. The PKCS#12 package is protected by an HMAC, and **ssh-certtool** prompts you for a passphrase before creating the package.

File arguments containing private keys can be read in naked PKCS#8 format, in ssh2 PEM format, or in openSSH PEM format. If the key is protected by a passphrase, **ssh-certtool** prompts for the passphrase. *File* arguments containing certificates are recognized in both DER-encoded and PEM-encoded format.

By default, the individual private key and certificates are saved into the PKCS#12 output file using default PBE protection schemes. The default scheme for key encryption is `pbeWithSHA1And3-KeyTripleDES-CBC`. The default for safe encryption is `pbeWithSHA1And40BitRC2-CBC` format. You can use the `-z` option to configure different PBE protection schemes.

OPTIONS

Options are available in both a single-character form (such as `-o`) and a descriptive equivalent (`--option`). Single characters are shown here. To view the descriptive equivalents, use the `-h` command line option.

-b *bits*

Specifies the key size used for generated keys. The default for RSA keys is 2048 bits and for DSA keys is 1024 bits. The value for a DSA key must be an integral multiple of 64. This option is valid for PKCS#10 file creation only.

-d *debug_level*

Enables debug output. Use 1, 2, 3, or 99. (Values 4-98 are accepted, but are equivalent to 3.)

-h

Displays a brief summary of command options.

-n *algorithm*

Specifies the algorithm used for key generation. Possible values are "rsa" and "dsa". The default is "rsa". This option is valid for PKCS#10 file creation only.

-o *output_file_prefix*

Specifies the first portion of the filename for output files. You can include an absolute path to generate the file in a different location. The default is "output". (The filename suffix is generated based on the file type: the suffix for PKCS#10 files is .pkcs10, for PKCS#12 is .p12, and for private keys is .ssh2.)

-p *private_key*

Specifies a private key to use in a certificate request. This option is valid for PKCS#10 file creation only.

-P

Saves the private key with an empty passphrase. This option is valid for PKCS#10 file creation only.

--passphrase *passphrase*

Specifies a passphrase for the private key. This option is valid for PKCS#10 file creation only.

-V

Displays product name and version information and exits. If other options are specified on the command line, they are ignored.

-z *Key=Value*

Specifies certificate options for PKCS#10 requests, and encryption options for PKCS#12 packages.

For PKCS#10 requests, *key* must be either **DNS** or **Email** (case-insensitive). There should be no white space in this option, including before or after the equal sign, unless the value literally contains white space characters in its name. The DNS option sets the DNS Alt Name extension. The Email option sets the EMail Alt Name extension. These extensions are not marked as critical.

For PKCS#12 packages, *key* must be either **KeyPBE** or **SafePBE** (case-insensitive). There should be no whitespace in this option, including before or after the equal sign. KeyPBE sets the key encryption and hmac scheme. SafePBE sets the safe encryption and hmac scheme. Values are listed below. The default for **KeyPBE** is PBE-SHA1-3DES. The default for **SafePBE** is PBE-SHA1-RC2-40. The long names in parentheses are synonyms.

None

PBE-SHA1-RC4-128 (pbeWithSHA1And128BitRC4)

PBE-SHA1-RC4-40 (pbeWithSHA1And40BitRC4)

PBE-SHA1-3DES (pbeWithSHA1And3-KeyTripleDES-CBC)

PBE-SHA1-2DES (pbeWithSHA1And2-KeyTripleDES-CBC)

PBE-SHA1-RC2-128 (pbeWithSHA1And128BitRC2-CBC)

PBE-SHA1-DES (pbeWithSHA1AndDES-CBC)

PBE-SHA1-RC2-40 (pbeWithSHA1And40BitRC2-CBC)

PBE-MD2-RC2-64 (pbeWithMD2AndRC2-CBC)

PBE-MD5-RC2-64 (pbeWithMD5AndRC2-CBC)

EXAMPLES

To create a PKCS#10 request using a newly generated key:

```
ssh-certtool -n RSA -z DNS=steves.dns.server.com -z Email=steved@myorg.org
pkcs10 CN=steved,O=myorg.org,OU=rsit,C=US DigitalSignature,nonRepudiation
ServerAuth,ClientAuth
```

To create a PKCS#12 package file and specify encryption for the key and safe:

```
ssh-certtool -z keyPBE=default -z safePBE=PBE-SHA1-RC4-40 -ofile pkcs12
id_rsa.crt id_rsa
```


Glossary of Terms

A

authentication

The process of reliably determining the identity of a communicating party. Identity can be proven by something you know (such as a password), something you have (such as a private key or token), or something intrinsic about you (such as a fingerprint).

C

CA (certification authority)

A server, in a trusted organization, which issues digital certificates. The CA manages the issuance of new certificates and revokes certificates that are no longer valid for authentication. A CA may also delegate certificate issuance authority to one or more intermediate CAs creating a chain of trust. The highest level CA certificate is referred to as the trusted root.

cipher

A cipher is an encryption algorithm. The cipher you select determines which mathematical algorithm is used to obscure the data being sent after a successful Secure Shell connection has been established.

CRL (Certificate Revocation List)

A digitally signed list of certificates that have been revoked by the Certification Authority. Certificates identified in a CRL are no longer valid.

D

data integrity

The assurance that data has not been changed from its original source. Methods to preserve data integrity are designed to ensure that data has not been accidentally or maliciously modified, altered or destroyed.

digital certificate

An integral part of a PKI (Public Key Infrastructure). Digital certificates are issued by a certification authority (CA), which ensures the validity of the information in the certificate. Each certificate contains identifying information about the certificate owner, a copy of the certificate owner's public key (used for encrypting and decrypting messages and digital signatures), and a digital signature (generated by the CA based on the certificate contents). The digital signature is used by a recipient to verify that the certificate has not been tampered with and can be trusted.

digital signature

Used to confirm the authenticity and integrity of a transmitted message. Typically, the sender holds the private key of a public/private key pair and the recipient holds the public key. To create the signature, the sender computes a hash from the message, and then encrypts this value with its private key. The recipient decrypts the signature using the sender's public key, and independently computes the hash of the received message. If the decrypted and calculated values match, the recipient trusts that the sender holds the private key, and that the message has not been altered in transit.

E

encryption

Encryption is the process of scrambling data by use of a secret code or cipher so it is unreadable except by authorized users. Encrypted data is far more secure than unencrypted data.

G

GSSAPI (Generic Security Services Application Program Interface)

An application programming interface that provides programs with access to security services.

H

hash

Also called a message digest, a hash or hash value is a fixed-length number generated from variable-length digital data. The hash is substantially smaller than the original data, and is generated by a formula in such a way that it is statistically unlikely that some other data will produce the same hash value.

K

Kerberos

A protocol that uses a trusted third party to enable secure communications over a TCP/IP network. The protocol uses encrypted tickets rather than plain-text passwords for secure network authentication.

M

MAC (Message Authentication Code)

Used to verify that data is not changed in transit, a MAC is a hash created using an arbitrary-length packet of data and a shared secret key. The sending and receiving party compute the MAC independently for each packet of transferred data using the shared key and an agreed-upon algorithm. If the message has changed in transit, the hash values are different and the packet is rejected.

P

passphrase

A passphrase is similar to a password, except it can be a phrase with a series of words, punctuation, numbers, white space, or any string of characters. Passphrases improve security by limiting access to secure objects, such as private keys and/or a key agent.

PKCS

PKCS (Public Key Cryptography Standards) is a set of standards devised and published by RSA laboratories that enable compatibility among public key cryptography implementations. Different PKCS standards identify specifications for particular cryptographic uses. The following standards are supported in Reflection for Secure IT Windows Server:

- PKCS#10 is used for certificate requests to a Certificate Authority (CA). You can use the **ssh-certtool** utility to create PKCS#10 files.
- PKCS#12 is used for storage and transportation of certificates and associated private keys. Files in this format typically use a *.pfx or *.p12 extension. Reflection for Secure IT supports authentication using certificates and keys stored in this format.

PKI Services Manager Configuration File

- UNIX
/opt/attachmate/pkid/config/pki_config
- Windows XP, Windows Server 2003:
\\Documents and Settings\\all users\\Application
Data\\Attachmate\\ReflectionPKI\\config\\pki_config
- Windows 7, Windows Vista, Windows Server 2008:
\\ProgramData\\Attachmate\\ReflectionPKI\\config\\pki_config

PKI Services Manager data folder

- Windows XP, Windows Server 2003:
\\Documents and Settings\\all users\\Application
Data\\Attachmate\\ReflectionPKI\\
- Windows 7, Windows Vista, Windows Server 2008:
\\ProgramData\\Attachmate\\ReflectionPKI\\

PKI Services Manager Map File

- UNIX
/opt/attachmate/pkid/config/pki_mapfile
- Windows XP, Windows Server 2003:
\\Documents and Settings\\all users\\Application
Data\\Attachmate\\ReflectionPKI\\config\\pki_mapfile
- Windows 7, Windows Vista, Windows Server 2008:
\\ProgramData\\Attachmate\\ReflectionPKI\\config\\pki_mapfile

port forwarding

A way to redirect unsecured traffic through a secure SSH tunnel. Two types of port forwarding are available: local and remote. Local (also called outgoing) port forwarding sends outgoing data sent from a specified local port through the secure channel to a specified remote port. You can configure a client application to exchange data securely with a server by configuring the client to connect to the redirected port instead of directly to the computer running the associated server. Remote (also called incoming) port forwarding sends incoming data from a specified remote port through the secure channel to a specified local port.

public key/private key

Public keys and private keys are pairs of cryptographic keys that are used to encrypt or decrypt data. Data encrypted with the public key can only be decrypted with the private key; and data encrypted with the private key can only be decrypted with the public key.

R

Reflection for Secure IT configuration file

The default configuration file location is:

Windows Server 2008:

```
\ProgramData\Attachmate\RSecureServer\rsshd_config.xml
```

Windows Server 2003:

```
\Documents and Settings\all users\Application  
Data\Attachmate\RSecureServer\rsshd_config.xml
```

Reflection for Secure IT data folder

The default data folder location is:

Windows Server 2008:

```
\ProgramData\Attachmate\RSecureServer\
```

Windows Server 2003:

```
\Documents and Settings\all users\Application  
Data\Attachmate\RSecureServer\
```

Reflection for Secure IT migration log file

Windows Server 2008:

```
\ProgramData\Attachmate\RSecureServer\Logs\migration.log
```

Windows Server 2003:

```
\Documents and Settings\all users\Application  
Data\Attachmate\RSecureServer\Logs\migration.log
```

regular expression

Often abbreviated as *regex*, a regular expression is a string of characters that describes one or more matching strings. Within a regular expression, some characters have a predefined meaning that determines what qualifies as a match. For example, the regular expression "t.*t" matches any word that starts and ends in the letter *t*, while the regular expression "text" matches only itself.

S**SCP1**

An early implementation of the SCP protocol used by OpenSSH. This protocol does not use the SFTP subsystem; it executes an **rcp** command through the secure channel.

SCP2

A file transfer implementation that uses the SFTP subsystem. SCP2 is useful for scripted file transfer.

Secure Shell

A protocol for securely logging onto a remote computer and executing commands. It provides a secure alternative to Telnet, FTP, rlogin, or rsh. Secure Shell connections require both server and user authentication, and all communications pass between hosts over an encrypted communication channel. You can also use Secure Shell connections to forward X11 sessions or specified TCP/IP ports through the secure tunnel.

SFTP

An interactive file transfer client that uses the sftp subsystem. SFTP transfer commands can also be used in batch files for automated transfers.

T**trust anchor**

A certificate that can be used as the final trust point in a certificate chain of trust. Note: PKI Services Manager validates certificates using only those trust anchors that have been explicitly configured for use by PKI Services Manager. You can configure a trust anchor using a root CA certificate, an intermediate CA certificate, or a self-signed certificates (one which can only validate itself).

U**UTC (Universal Time, Coordinated)**

A high-precision time standard. When describing time zones, UTC refers to the time kept on the Greenwich meridian (longitude zero), also known as *Greenwich Mean Time*. UTC times are generally given in terms of a 24-hour clock.

W

Windows home folder

The home folder is configurable by the Windows system administrator. When no home folder is configured (the default), the home folder is the same as the User profile. The default User profile is:

- Windows 7, Windows Vista, Windows Server 2008:
`\Users\username\`
- Windows XP, Windows Server 2003:
`\Documents and Settings\username\`

Windows user profile folder

The user profile folder is configurable by the Windows system administrator. The default is:

- Windows 7, Windows Vista, Windows Server 2008:
`\Users\username\`
- Windows XP, Windows Server 2003:
`\Documents and Settings\username\`

Index

A

- access control
 - access overview • 111
 - client host access • 114
 - group access • 118
 - IP blocking • 56
 - Permissions pane • 125
 - user access • 122
- asterisk • 129
- authentication
 - Authentication pane • 56
 - certificate • 48
 - Kerberos (GSSAPI) • 51
 - public key • 42
 - server • 41
 - user • 55

B

- banner message • 23
- basics
 - getting help • 19
 - getting started • 17
 - understanding Secure Shell • 19

C

- cache passwords
 - Credential Cache pane • 87
 - set up password caching • 86
- certificate authentication
 - about user authentication • 65
 - Certificates pane • 70
 - configure server authentication • 49
 - configure user authentication • 67
 - introduction (server authentication) • 48
- certificates
 - ssh-certtool • 196
- chroot • 103
- Client Host Access Control dialog box • 116
- Client Host Access Control Pane • 115
- clusters
 - configure a cluster • 30

- overview • 29
- cmd.exe • 125
- command line utilities
 - rssh • 191
 - settings • 125
- command shell
 - command shell access • 125
- configuration file
 - changing the data folder • 26
 - default location • 205
- credential cache
 - Add/Edit Credential • 91
 - best practices • 83
 - Credential Cache pane • 87
 - Filters dialog box • 94
 - how to record and use cached passwords • 86
 - overview • 82
 - Select Credential dialog box • 92

D

- data encryption • 35
- data folder
 - changing • 26
 - default data folder • 205
 - Set Data Folder dialog box • 33
- Debug Logging pane • 143
- default settings • 22
- deploying
 - administrative installation • 16
- directory paths
 - pattern strings • 165
- Domain Access pane • 90
- domain name string • 165
- drive mapping
 - configure mapped drives for terminal sessions • 95
 - Mapped Drives pane • 97
 - Mapped Drives Settings dialog box • 98

E

- encryption
 - data encryption • 35
 - Encryption pane • 37
- Event Logging pane • 142
- Event Viewer • 141
- Export Public Key dialog box • 47

F

- file transfer
 - overview • 99
 - permissions • 125
 - resume • 106
 - SFTP directories • 107
- Files Used by Reflection for Secure IT • 150
- FIPS
 - configure • 37
 - defined • 36
 - effects • 36

G

- General pane • 23
- Generate Host Private Key dialog box • 46
- group settings
 - group access • 118
 - Group Access Control dialog box • 121
 - Group Access Control pane • 120
 - Group Configuration dialog box • 134
 - Group Configuration pane • 134
 - group-specific settings • 133
 - revert settings to inherited values • 138
 - troubleshooting • 146
- GSSAPI
 - configure client authentication • 77
 - configure mutual authentication • 52
 - GSSAPI / Kerberos V5 Pane • 79
 - Kerberos authentication • 51

H

- help
 - troubleshooting • 147
 - using the help • 19
- home directory string • 165
- host key
 - configure server authentication • 43
 - Identity tab • 44

I

- installation
 - administrative installation • 16
 - install the server • 11
 - upgrade • 14
- IP address
 - Network Binding dialog box • 25
- IP blocking • 56

K

- Kerberos
 - configure client authentication • 77
 - configure mutual authentication • 52
 - GSSAPI / Kerberos V5 Pane • 79
 - Kerberos authentication • 51
- key authentication
 - configure server authentication • 43
 - configure UNIX client authentication • 62
 - configure Windows client authentication • 61
 - introduction • 42
- key exchange
 - introduction • 38
 - Key Exchange pane • 38
- keyboard access • 167
- keys
 - Identity tab • 44
 - ssh-keygen • 193

L

- logging
 - Custom Log Events dialog box • 145
 - debug log file • 143
 - Debug Logging pane • 143
 - Event Logging pane • 142
 - Windows Event Viewer • 141
- login directory
 - login directory • 100
 - terminal sessions • 125

M

- MACS
 - overview • 36

mapped drives

- configure mapped drives for terminal sessions • 95

- Mapped Drives pane • 97

- Mapped Drives Settings dialog box • 98

- maximum number of connections • 23

migrating settings

- install over earlier version • 14

- table of migrated settings • 154

MSCS clustering

- configure a cluster • 30

- overview • 29

N

- Network Binding dialog box • 25

- Network pane • 24

P

passwords

- cache passwords • 86

- configure user authentication • 58

- Password pane • 60

- pattern strings • 165

- Permissions pane • 125

- pki-mapfile • 178

port

- change the port • 24

- Network Binding dialog box • 25

port forwarding

- disable • 140

- overview • 139

- profile directory string • 165

public key

- configure host key • 44

- export host key • 47

- generate host key • 46

public key authentication

- configure server authentication • 43

- configure UNIX client authentication • 62

- configure Windows client authentication • 61

- how to record and use cached passwords • 86

- introduction • 42

- Public Key pane • 63

R

RADIUS authentication

- configure RADIUS authentication • 75

- RADIUS pane • 76

- RADIUS Server dialog box • 77

- regular expression syntax • 152

- rekey interval • 38

- root directories • 103

RSA SecurID authentication

- Configure SecurID authentication • 73

- RSA SecurID pane • 74

- rsshd command line utility • 191

S

Secure Shell

- understanding Secure Shell • 19

- session timeout • 23

- Setting is no longer inherited • 129

SFTP

- configure • 101

- permissions • 125

- resume • 106

- SFTP Directories pane • 107

- user login directory • 100

- ssh-keygen • 193

- start the server • 18

- Status tab • 22

- stop the server • 18

subconfiguration

- client host access • 114

- group access • 118

- group-specific settings • 133

- overview • 129

- revert settings to inherited values • 138

- user access • 122

- user-specific settings • 135

T

terminal shell

- command shell access • 125

- login directory • 125

- settings • 125

troubleshooting

- command shell access • 125

- group settings • 146
- help • 147
- logging • 141
- managing system resources • 146
- manual host key migration • 164

tunneling

- disable • 140
- overview • 139

U

upgrade

- install over earlier version • 14
- migrated settings • 154

user login string • 165

user settings

- revert settings to inherited values
 - 138
- user access • 122
- User Access Control dialog box • 124
- User Access control pane • 123
- User Configuration dialog box • 136
- User Configuration pane • 136
- user-specific settings • 135

V

virtual directories

- Accessible Directory Settings dialog box • 109
- configure • 101
- overview • 103
- pattern strings • 165

W

winpki • 168, 172