SSH Communications
Tectia 6.4.5

RSA SecurID Ready Implementation Guide

Last Modified: December 16, 2013

Partner Information

Product Information	
Partner Name	SSH Communications Security Corp
Web Site	www.ssh.com
Product Name	Tectia
Version & Platform	6.4.5
Product Description	SSH Tectia is the leading end-to-end communications security solution for the enterprise. The SSH Tectia solution is based on the SSH Secure Shell and SSH's other industry leading technologies used by millions worldwide. SSH Tectia enables secure system administration, secure file transfer and secure application connectivity with centralized management throughout the internal and external network. SSH Tectia provides transparent strong encryption and authentication and easily integrates into heterogeneous, multi-platform environments.

ssh
Securing the path to your information assets

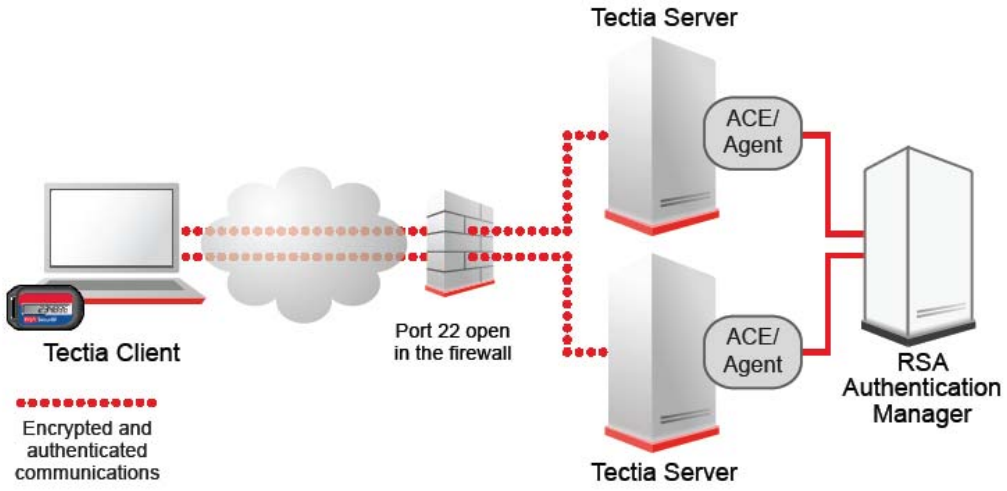
Solution Summary

SSH Tectia Client and Server form an enterprise-class Secure Shell solution for securing system administration, file transfer, and application connectivity in heterogeneous enterprise networks. SSH Tectia Client and Server are based on the IETF standard Secure Shell (version 2) protocol.

SSH Tectia Client and Server support using RSA SecurID for two-factor authentication. SSH Tectia Client forwards the SecurID passwords using a method called keyboard-interactive. It enables implementation of new authentication schemes based on keyboard interaction without the need to modify the client side.

SSH Tectia Server includes support for RSA Authentication Agent API. The RSA Authentication Agent software is installed on the same server. The Agent connects to the RSA Authentication Manager that performs the user identity validation against the one-time password generated by the SecurID token.

RSA Authentication Manager supported features	
Tectia 6.4.5	
RSA SecurID Authentication via Native RSA SecurID Protocol	Yes
RSA SecurID Authentication via RADIUS Protocol	No
On-Demand Authentication via Native SecurID Protocol	Yes
On-Demand Authentication via RADIUS Protocol	No
Risk-Based Authentication	No
Risk-Based Authentication with Single Sign-On	No
RSA Authentication Manager Replica Support	Yes
Secondary RADIUS Server Support	No
RSA SecurID Software Token Automation	No
RSA SecurID SD800 Token Automation	No
RSA SecurID Protection of Administrative Interface	No



Authentication Agent Configuration

Authentication Agents are records in the RSA Authentication Manager database that contain information about the systems for which RSA SecurID authentication is provided. All RSA SecurID-enabled systems require corresponding Authentication Agents. Authentication Agents are managed using the RSA Security Console.

The following information is required to create an Authentication Agent:

- Hostname
- IP Addresses for network interfaces

Set the Agent Type to "Standard Agent" when adding the Authentication Agent. This setting is used by the RSA Authentication Manager to determine how communication with SSH Tectia Server will occur.

Note: Hostnames within the RSA Authentication Manager / RSA SecurID Appliance must resolve to valid IP addresses on the local network.

RSA SecurID files

RSA SecurID Authentication Files	
Files	**Location**
sdconf.rec	Same as aceclnt.dll or libaceclnt.so
Node Secret	Same as aceclnt.dll or libaceclnt.so
sdstatus.12	Same as aceclnt.dll or libaceclnt.so
sdopts.rec	Same as aceclnt.dll or libaceclnt.so

Note: The appendix of this document contains more detailed information regarding these files.

Partner Product Configuration

Before You Begin

This section provides instructions for configuring SSH Tectia Server with RSA SecurID Authentication. This document is not intended to suggest optimum installations or configurations.

It is assumed that the reader has both working knowledge of all products involved, and the ability to perform the tasks outlined in this section. Administrators should have access to the product documentation for all products in order to install the required components.

All SSH Tectia Server components must be installed and working prior to the integration. Perform the necessary tests to confirm that this is true before proceeding.

Server Configuration

This section contains the following instructions:

- SSH Tectia Server configuration on Windows
- SSH Tectia Server configuration on UNIX
- SSH Tectia Client configuration on Windows
- SSH Tectia Client configuration on UNIX

Depending on your client and server platforms, carry out the steps on each client and server you wish to enable SecurID authentication on.

SSH Tectia Server Configuration

The keyboard-interactive authentication with the SecurID submethod is used to enable RSA SecurID authentication on SSH Tectia Server.

Important: Create an identical user record on both the RSA Authentication Manager and the operating system (local or domain).

Windows Server

To enable SecurID authentication on SSH Tectia Server on Windows, do the following steps:

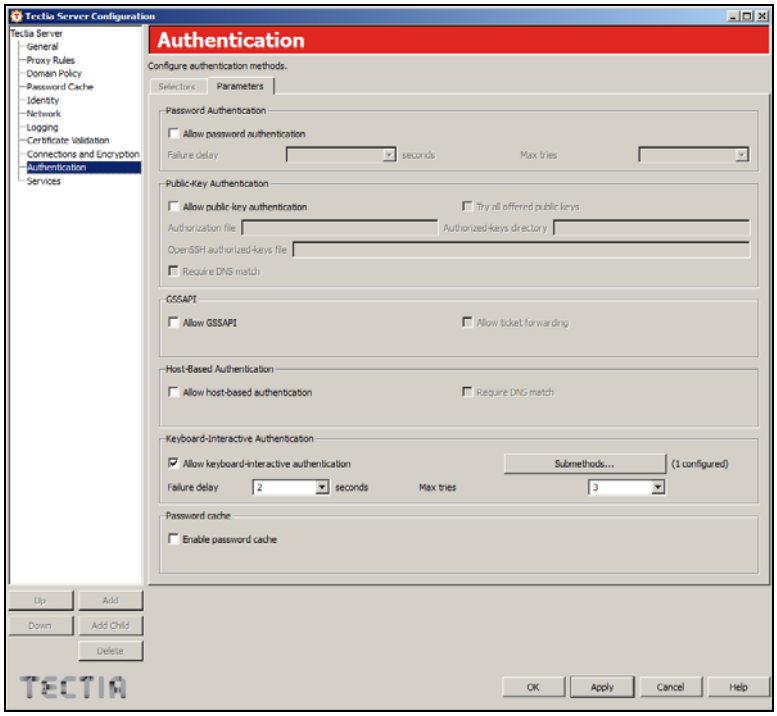
1. Perform a standard installation of the RSA ACE Client. Reference the RSA product documentation for details.

   Note: Installing SSH Tectia on Windows 64 bit OS requires installing the 32 bit RSA Authentication Agent on a Windows 32 bit system. Once installed copy the aceclnt.dll and sdmsg.dll from the "C:\Program Files\Common Files\RSA Shared\Auth Data" directory and place the files on the Windows 64 bit Server in the "C:\Program Files (x86)\SSH Communications Security\SSH Tectia\SSH Tectia Server" directory with the sdconf.rec file from the RSA Authentication Manager.

   Add "C:\Program Files (x86)\SSH Communications Security\SSH Tectia\SSH Tectia Server " in the System Path of the Windows Environment Variables.

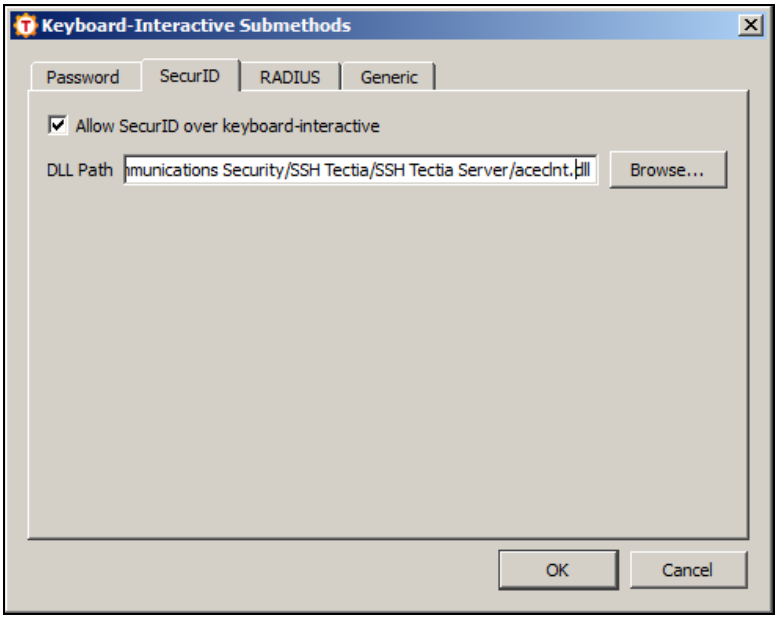
2. Launch the SSH Tectia Server Configuration tool from the **Start menu, Programs > SSH Tectia Server > SSH Tectia Server Configuration**.

3. From the tree view on the left, select **Authentication** and **uncheck** all authentication methods except Keyboard-Interactive Authentication. Click **Submethods**.

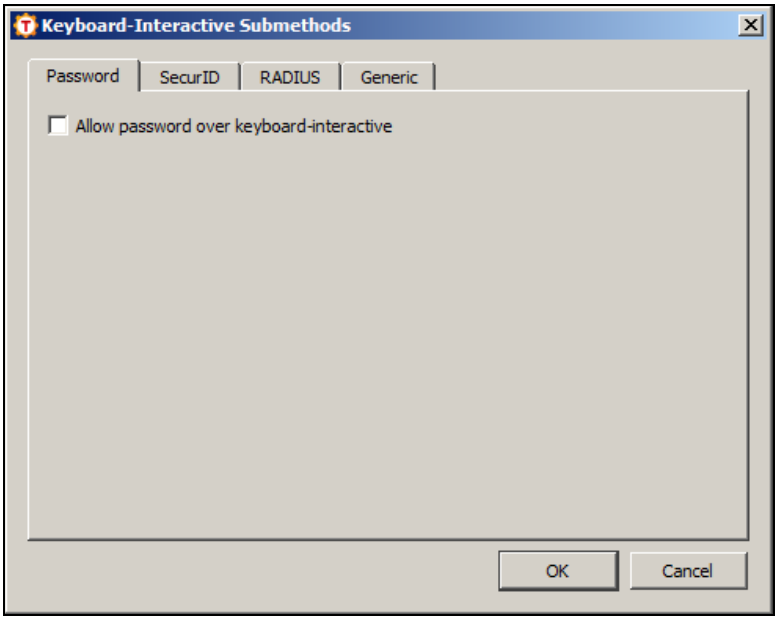


4. Select the **SecurID** Tab and check **Allow SecurID over keyboard-interactive**. Select **Browse** and locate the folder where aceclnt.dll is stored.

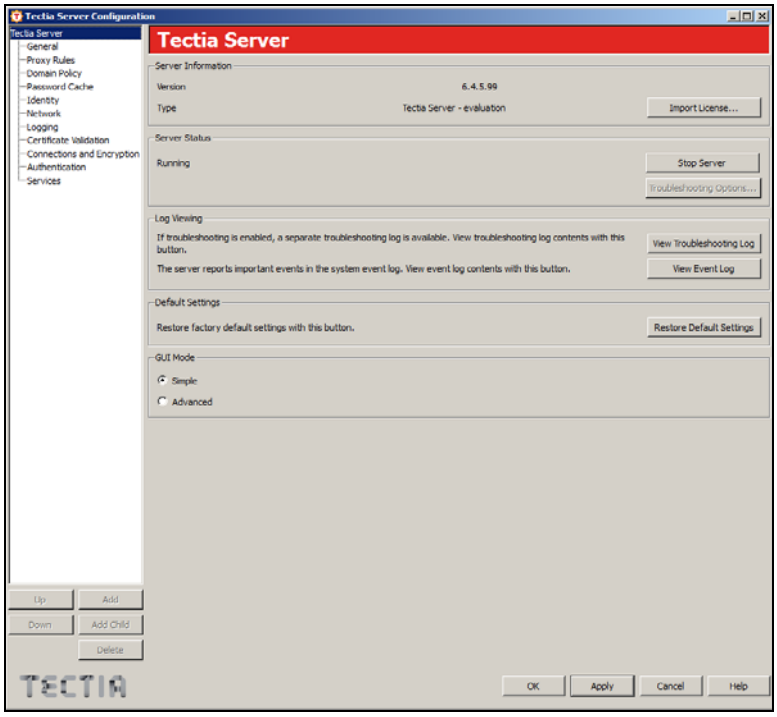
> **Note: See appendix for the location of the aceclnt.dll.**



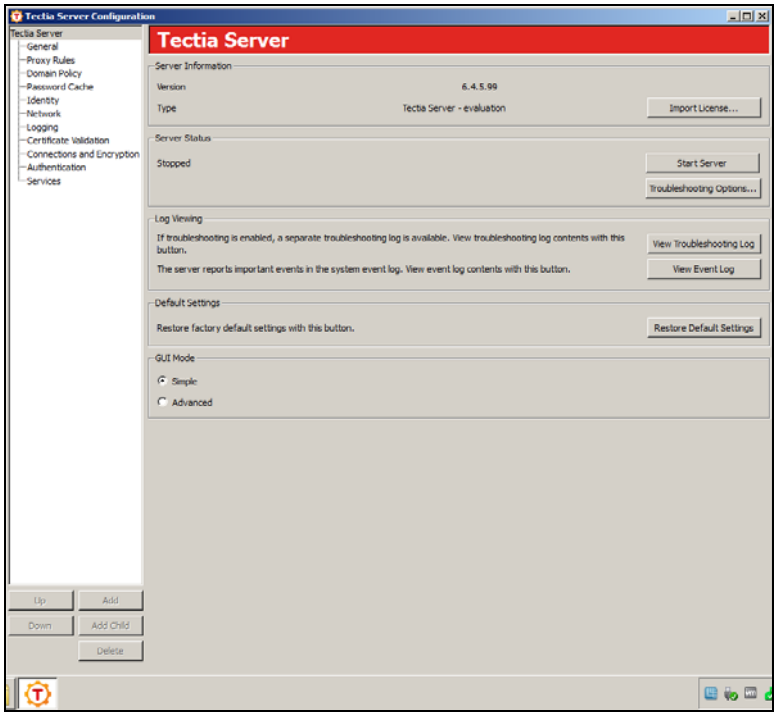
5. Select the **Password Tab** and uncheck **Allow password over keyboard-interactive**. Select **OK**.



6. From the tree view on the left, select **Tectia Server** and then select **Stop Server**.



7. Select **Start Server**.



UNIX Server

To enable SecurID authentication on SSH Tectia Server on UNIX, do the following steps:

1. Install the Local Authentication Client feature of the RSA Authentication Agent. On UNIX platforms, SSH Tectia Server has been tested with RSA Authentication Agent for UNIX 5.2 and RSA Authentication Agent for PAM 5.3.4.

> **Note: For the SecurID authentication to work with SSH Tectia Server on UNIX, the RSA ACE/Agent `libaceclnt.so` library has to be available in the `/usr/lib` directory (alternatively `/user/ace/lib` or `/opt/ace/lib`).**

2. Edit the `/etc/ssh2/ssh-server-config.xml` configuration file to offer keyboard-interactive authentication with the SecurID submethod:

```
<authentication-methods>
  <authentication action="allow">
    <auth-keyboard-interactive>
      <submethod-securid />
    </auth-keyboard-interactive>
  </authentication>
</authentication-methods>
```

   Also other authentication methods can be allowed. To disable other authentication methods, remove the corresponding entries from the configuration file.

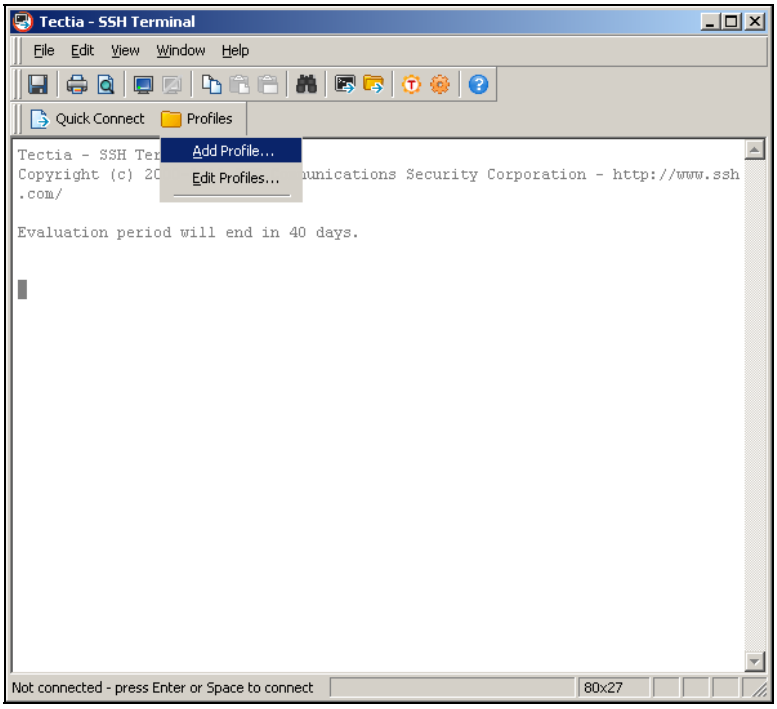
3. Restart SSH Tectia Server, for example, on Linux and Solaris:

```
# /etc/init.d/ssh-server-g3 restart
```

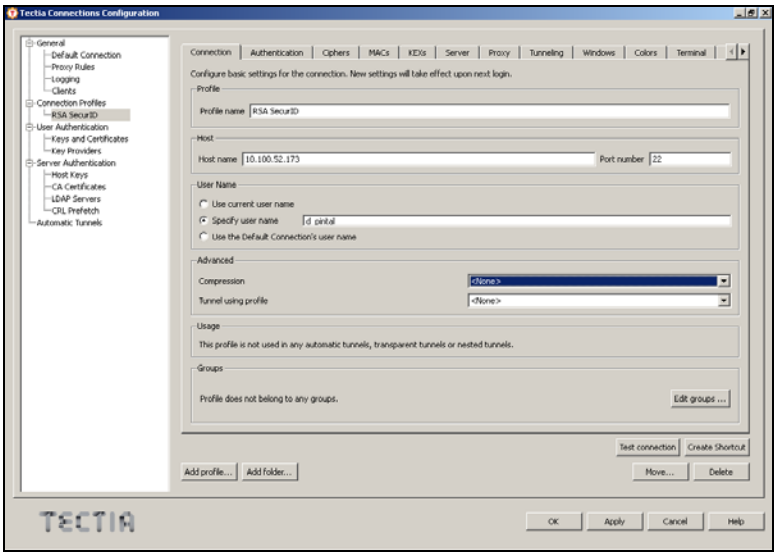
The server is now ready to accept connections using SecurID authentication.

SSH Tectia Client Configuration

1. Open the Tectia SSH Terminal and from the Profiles drop down select **Add Profile**.



2. Enter the Profile Name as **RSA SecurID**, Enter the Host name or IP Address of your SSH Tectia Server and Specify the **username**.



UNIX Client

1. Edit the `$HOME/.ssh2/ssh-broker-config.xml` configuration file to enable keyboard-interactive authentication (it is enabled by default):

```
<default-settings>
...
  <authentication-methods>
    <authentication-method name="keyboard-interactive" />
  ...
  </authentication-methods>
...
</default-settings>
```

Also other authentication methods can be listed. To disable other authentication methods, remove the corresponding entries from the configuration file. List the authentication methods in the order you want to attempt them.

2. SSH Tectia Client can have connection profiles that define customized connection settings for different Secure Shell servers. To customize the authentication settings in the connection profiles, make the appropriate changes in the configuration file under the `<profile>` element:

```
<profiles>
  <profile name="server1" id="id1" host="10.100.52.173"
           port="22" user="%username%">
    <authentication-methods>
      <authentication-method name="keyboard-interactive" />
    </authentication-methods>
  </profile>
...
</profiles>
```

The client is now ready to use SecurID authentication.

To test the connection, connect with `sshg3` to the server, for example:

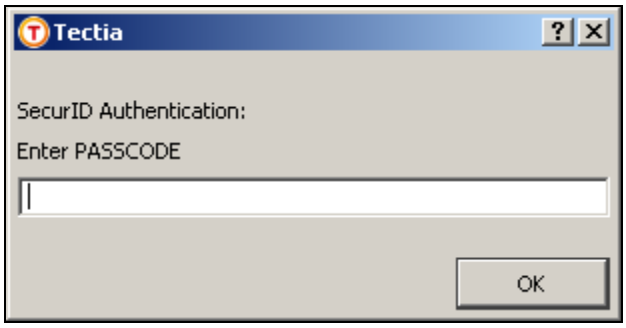
```
$ sshg3 securid_user@server1
```

Once the SSH Tectia Server accepts keyboard-interactive as the authentication method, SSH Tectia Client will prompt the user for the RSA SecurID PASSCODE.

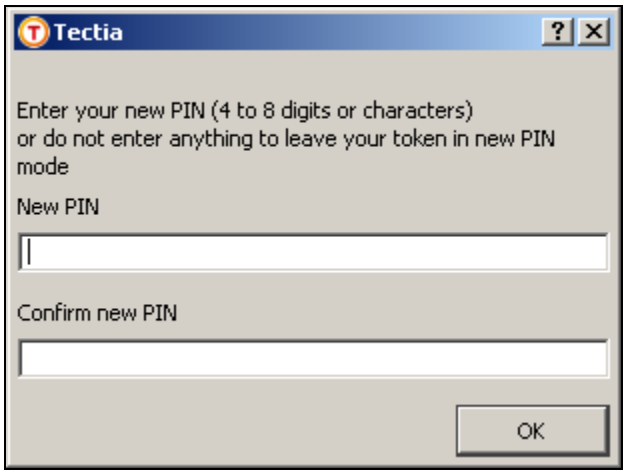
```
Keyboard-interactive:
SecurID Authentication:
Enter PASSCODE:
```

RSA SecurID Login Screens

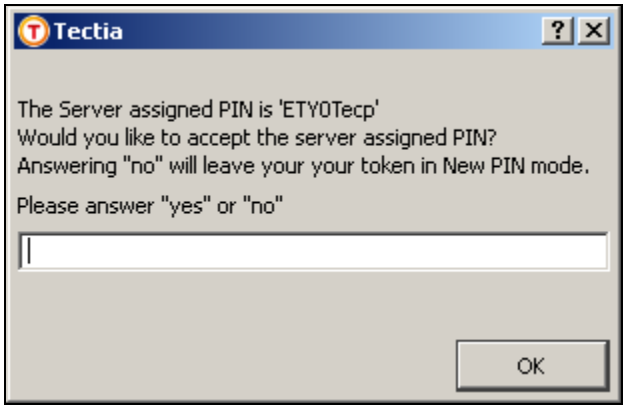
Login screen:



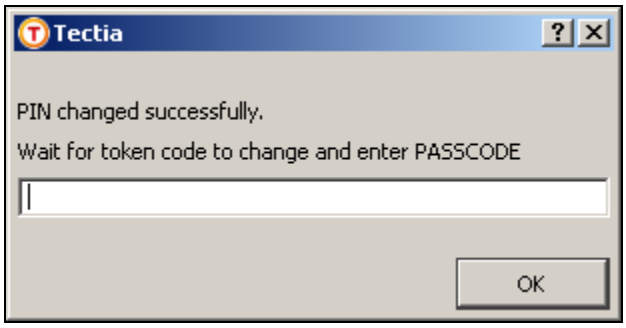
User-defined New PIN:



System-generated New PIN:



Next Tokencode:



Certification Checklist for RSA Authentication Manager

Date Tested: December 16, 2013

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA Authentication Manager	8.0	Virtual Appliance
RSA Authentication Agent	7.2.1	Windows 2008 R2 x64
SSH Tectia	6.4.5	Windows 7 SP2 x64

Mandatory Functionality			
RSA Native Protocol		**RADIUS Protocol**	
New PIN Mode			
Force Authentication After New PIN	✓	Force Authentication After New PIN	N/A
System Generated PIN	✓	System Generated PIN	N/A
User Defined (4-8 Alphanumeric)	✓	User Defined (4-8 Alphanumeric)	N/A
User Defined (5-7 Numeric)	✓	User Defined (5-7 Numeric)	N/A
Deny 4 and 8 Digit PIN	✓	Deny 4 and 8 Digit PIN	N/A
Deny Alphanumeric PIN	✓	Deny Alphanumeric PIN	N/A
Deny PIN Reuse	✓	Deny PIN Reuse	N/A
Passcode			
16-Digit Passcode	✓	16-Digit Passcode	N/A
4-Digit Fixed Passcode	✓	4-Digit Fixed Passcode	N/A
Next Tokencode Mode			
Next Tokencode Mode	✓	Next Tokencode Mode	N/A
On-Demand Authentication			
On-Demand Authentication	✓	On-Demand Authentication	N/A
On-Demand New PIN	✓	On-Demand New PIN	N/A
Load Balancing / Reliability Testing			
Failover (3-10 Replicas)	✓	Failover	N/A
No RSA Authentication Manager	✓	No RSA Authentication Manager	N/A

DRP / PAR

✓ = Pass ✗ = Fail N/A = Not Applicable to Integration

Appendix

Partner Integration Details	
RSA SecurID API	8.1.2
RSA Authentication Agent Type	Standard Agent
RSA SecurID User Specification	Default Method
Display RSA Server Info	No
Perform Test Authentication	Yes
Agent Tracing	Yes

SSH Tectia is a 32bit application and supports a variety of OSes. When deploying to a Windows 32 bit OS the RSA Authentication Agent is installed normally. If installing SSH Tectia to a Windows 64 bit OS the 32 bit RSA Authentication Agent cannot be installed. As a result to properly configured the Windows 64 bit server, two libraries (aceclnt.dll and sdmsg.dll) from the RSA Authentication Agent are required. In addition the location of these files must be defined within the Variable Path of the Windows Environment System Variables.

Recommended Locations for aceclnt.dll and libaceclnt.so

Windows 32bit OS

```
C:\Program Files\Common Files\RSA Shared\Auth Data
```

Windows 64bit OS

```
C:\Program Files (x86)\SSH Communications Security\SSH Tectia\SSH Tectia Server
```

Linux/UNIX

```
/usr/lib (alternatively /user/ace/lib or /opt/ace/lib)
```

This section is provided to show an administrator how to load, remove, or update the sdopts.rec, sdstatus.12 and Node Secret file if it was not previously documented under the Partner Authentication Agent Configuration section. It is also provided to list any technologies or terms specific to the Partner product that may not be viewed as common knowledge. If a testing utility has been added to the product so that you can test RSA SecurID authentications from the product then add a note on how to get to and use the utility.

Node Secret:

Delete securid from the same folder where aceclnt.dll or libaceclnt.so are installed.

sdconf.rec:

Delete sdconf.rec from the same folder where aceclnt.dll or libaceclnt.so are installed.

sdopts.rec:

Delete sdopts.rec from the same folder where aceclnt.dll or libaceclnt.so are installed.

sdstatus.12:

Delete sdconf.rec from the same folder where aceclnt.dll or libaceclnt.so are installed.

Windows Agent Tracing:

Using Regedit locate the HKEY_LOCAL_MACHINE\Software\SDTI\ACECLIENT key and create 2 DWORD values: **tracelevel** and **tracedest.**

> **Note: SDTI\ACECLIENT and all sub values will need to be created when setting up for Windows 64 bit OSes.**

The value tracelevel specifies the verbosity and the categories of messages produced by the code. The value tracedest controls the output destination of the trace messages.

tracedest VALUES:

```
SDITRACE_EVENT_LOG  0x00000001   // messages to event log
SDITRACE_CONSOLE    0x00000002   // messages to console
SDITRACE_LOGFILE    0x00000004   // messages to logfile (aceclient.log)
SDITRACE_DEBUGGER   0x00000008   // messages to debugger output
SDITRACE_NOFILELINE 0x80000000   // no file and line information
```

The SDITRACE_NOFILELINE value can be combined with any of the other values to stop the display of file and line number information. The logfile is SYSTEMROOT%\ACECLIENT.LOG but can be changed by creating a **REG_SZ:tracefile** value and specifying the file pathname.

tracelevel VALUES:

```
SDITRACEING_OFF     0x000000000  // All messages off
SDITRACEING_ON      0x000000001  // All messages marked with this level on
SDITRACEING_ENTRY   0x000000002  // All entrypoints use this
SDITRACEING_EXIT    0x000000004  // All function returns use this
SDITRACEING_FLOW    0x000000008  // All logic flow control use this (ifs)
SDITRACEING_GRP1    0x000000010  // Old SDITRACE macros use this (see dbglib.h)
```

The hex value 0xF gives the complete set of tracing. The values can be combined to produce multiple sets of trace messages.

> **Note: Using the SDITRACE_CONSOLE value can cause the service applications to access violate during logoff. Use only for real time debugging situations.**