CHAPTER 1

Enterprise Deployment

This chapter contains the following sections:

- Introduction, page 1-1
- Distribution Package, page 1-2

Introduction

The Cisco Secure Services Client (SSC) is an 802.1X authentication supplicant for creating secure wired and wireless connections. SSC also has a user interface for displaying status and accepting commands from a user. It allows your computer to connect to a network that is protected by the IEEE 802.1X security protocol. Only after successful client-server authentication will the port access control on the 802.1X-enabled access device (the wireless access point or the wired Ethernet switch) allow end-user connectivity to the network.

SSC has two basic versions:

- The out-of-the-box version

  SSC as downloaded from cisco.com is not configured. It is intended for use by an IT organization that is responsible for configuring and deploying a derived, end-user version. This deployed version is appropriate for use by the various enterprise departments and organizations that you support. As the IT Administrator you have control over the user experience and the end-user's allowed choices and configuration options. The out-of-the-box version has a fully open policy that allows access to most features and requires configuring a network when initially started. However, only through a deployed distribution package file, that is, a SSC configuration file, does the IT Administrator have full access to all settings and network configurations.

  There are two configurations for the out-of-the-box version:

  - Default download package—configured with a nonexpiring, wired-only license.
  - Re-licensed package—adds a trial, wireless license.
    (Also available for download from cisco.com on the SSC page.)
    See "Activating the Client" in the companion *Cisco Secure Services Client User Guide*.

    Once activated with the wireless trial license, you are able to:

    (1) Evaluate wireless functionality for 90 days, via the temporary license.

    (2) Permanently license the product for both wired and wireless functionality.

- The deployed end-user version

The deployed end-user version is pre-configured with a distribution package description, possibly with a restricted feature set, and deployed by you the IT/System Administrator. It most likely contains one or more pre-defined enterprise networks that allow instant connection to your enterprise networks. Two types of end-user interfaces are available as follows:

- The configurable end-user version

  This version allows your end-users to create new network profiles within the scope of your policy. It is an excellent choice for end-stations that will move out of the enterprise network to home or travel networks.

- The preset end-user version

  This version contains only your pre-defined network profiles that allow instant connection to your enterprise networks. It is an excellent choice for end-stations that will encounter only enterprise networks that you control.

Note The out-of-the-box default wired SSC supports:

- Wired (802.3) network adapters
- EAP methods: EAP-FAST with EAP-MSCHAPv2, EAP-GTC, EAP-TLS
- Smartcard provided credentials
- Cisco Trust Agent (CTA) processing when CTA is also installed

The trial license supports:

- Wireless (802.11) network adapters
- WPA2/802.11i protocols
- EAP methods: LEAP, EAP-PEAP, EAP-TTLS, EAP-MD5

Supported Operating System Environments

The supported operating system environments are:

- XP Professional (SP1, SP2), 2K (SP4), Win2K Servers (SP4), Win2003 Server
- Novell Client version 4.91 SP1 with Hotfix TID2972711

Note By omission, other editions of Windows XP such as Home, Media Center, Tablet PC, Professional x64 and so forth, are not supported.

Distribution Package

The distribution package defines how an individual end-user SSC operates and creates connections. A distribution package contains the following functional blocks:

- License

  The deployed end-user SSC initially requires the enterprise license that you obtained from Cisco Systems. This will replace the trial license built into the out-of-the-box version.

- Policy
  - User control policy

    Sets the deployed type and network media support.
  - Network policy

    Sets the limitations on the types and capabilities of all supported networks.
- Connection Settings

  Configures the global operational aspects of making network connections.
- Networks

  Contains a single or a set of network profile descriptions. A network profile defines the specific properties and operational behavior of a single network. This profile includes the following characteristics:
  - The user-friendly name of the network.
  - Network access media (wired, Wi-Fi) and adapter details used for the network connection.
  - Definition of the security class (open, shared key, authenticating) of the network.
  - Definition of the connection context (machine only, user only, machine and user) for the network.
  - Wi-Fi Association and Encryption method (Wi-Fi network).
  - Authentication methods supported and properties (authenticating network).
  - Static keys, if applicable (non-authenticating network).
  - Definition of types and source of credentials (authenticating network).
  - Definition of trusted servers (authenticating network) and support for deploying Certificate Authority (CA) certificates and manual provisioning of EAP-FAST Protected Access Credentials (PACs).

  Networks defined as part of the distribution package are locked; that is, the end-user is not able to edit the configuration settings.

The major steps that must take place to tailor the SSC to the desired enterprise environment are:

1. Creation—The administrator creates a distribution package file. A single distribution package file may contain configuration descriptions for more than one network. See "Distribution Package Creation" for complete details on the format, structure and contents of the distribution package.
2. Deployment—The administrator packages the application and/or the distribution package file and deploys to the end station. See section "Distribution Package Deployment" for details on deployment options and instructions.
3. Introduction—The SSC detects and uses the distribution package file. This step is automatic and does not require any administrator intervention. Shortly after the deployment step, the existence of the new distribution package file is detected. It is then processed for validity and, if valid, the SSC reconfigures itself accordingly.

Distribution Package Utilities

All of the utility tools and support files needed for the creation and deployment of a distribution package are contained in a single packaged file, SSCAdminUtils_{version}.zip. The individual items are introduced and described in the remainder of this chapter.

You can download the utility package online at the Cisco SSC download page. Go to SSC product support at:

http://www.cisco.com/en/US/products/ps7034/tsd_products_support_series_home.html

and click on **Download Software** and follow the **wireless software** links to the SSC download page.

Distribution Package Creation

Distribution Package Schema

SSC utilizes the XML format for the distribution package file. The overall structure of a specific .xml distribution package (configuration) file is defined by the SSC distribution package schema, distributionPackage.xsd.

The SSC distribution package schema is a standard W3C XML Schema compliant document used for describing and constraining the content of any .xml configuration file. It is assumed that the user of this document is familiar with the syntax of the W3C XML Schema specification and an instantiated XML output.

Schema Properties

The schema has the following aspects:

- Any distribution package instance XML file is a readable text file that helps the reader to fully understand the end-user configuration. To support user readability the schema has the following characteristics:
  - Each configuration setting is represented by a specific schema element.
  - Configuration settings are conveyed by the existence of an optional element or a value of an element.
  - The use of schema attributes is reserved for clarifying a configuration setting.
- The definition of a network is a hierarchical decision tree structure. The schema walks you through the tree based on your choices as you proceed. Traversing the tree automatically narrows down the set of configurable parameters to those that are of concern for your particular type of network. Additionally, this automatically refines the set of values allowed for a given configuration parameter. For example, in a wireless network one needs to configure an association mode for the connection. But the set of allowed values if you choose an authenticating network is different than if you choose a shared network. The basic order in which decisions are made is as follows:

  For all networks:

  1. Choosing connection media (wired or wireless) for the network.
  2. Choosing security class (open, shared key, authenticating) for the network.
  3. Choosing connection context (machine only, user only, machine and user) for the network.

  For an authenticating network the decision tree continues:

4. Choosing credential type and collection method.
5. Choosing authentication method(s).

Schema Validation:

Although the schema includes enumeration values it does not explicitly specify all of the allowed uses and combinations of elements, nor requirements for non-enumerated strings. Those details are covered by a set of Business Rules (which are detailed later in this document in Chapter 2, “Schema Elements”).

A generated .xml distribution package file must, therefore, satisfy the following criteria in order to be accepted by the SSC.

- The .xml file must be valid with respect to the syntactical requirements of the SSC distribution package schema.
- The .xml file must be valid with respect to the element relationship requirements of the schema Business Rules.

Distribution Package Creation Steps

Follow these steps to create a distribution package file.

Step 1 Generate the descriptive .xml distribution package file as specified by the SSC schema and this document (Chapter 2, “Schema Elements”). Alternative methods for accomplishing this include:

- Use a commercially available XML editor that supports direct creation of an XML instance file from a schema. These tools provide some contextual help during the XML editing and helps you validate the instance file. Examples of such applications are:
  - XMLSpy by Altova
  - Stylus Studio by DataDirect Technologies
- Use any text editor and the detailed description of the schema structure and elements given in Chapter 2, “Schema Elements” to create an XML instance file either from scratch or by cut-and-paste from the included examples.
  - From scratch—Chapter 2, “Schema Elements” walks you through the schema and contains XML examples of element structure for reference.
  - Cut and paste—Appendix B, “Distribution Package Examples” contains complete distribution package examples. Pick one from the list that best matches your network environment and edit it by referring to the details in Chapter 2, “Schema Elements”. The file, sscAdminGuideExXml.zip, also distributed in the SSCAdminUtils zip file, contains all of the examples as individual .xml files, for a convenient starting point and easy text editing.

Tip Text editing is greatly simplified by using a programming text editor that recognizes the syntax of the text language (in this case, XML). There are many such editors available commercially. Some support additional features such as automatic tag closing and element indentation cleanup.

Tip XML Syntax:

The syntax rules of XML are very simple. A few basic concepts are listed here:

- Each .xml file has a root element, in our case *configuration*, which servers as the container for the descriptive elements.
- All XML elements must have a closing tag.
- XML elements must be properly nested.
- XML tags are case sensitive.
- An element may contain child elements, content (text values) or attributes, in any combination.
- All attribute values must be quoted.
- White space is preserved. (This is important, for example, when entering specified enumerated content values. Avoid leading and trailing white space for enumerated and boolean values.)
- A comment is surrounded by the following syntax: <!-- your comment -->.

A specific .xml distribution package file (also known as an instance of the distribution package schema) is therefore constructed from the following building blocks:

```
<configuration>
  <childElement>with content</childElement>
  <elementWithAttr attr=”{value}”>
    <anotherChild>
       <!-- more hierachical elements -->
    </anotherChild>
  </elementWithAttr> <!--properly nested closing tag-->
  <emptyElement1></emptyElement1> <!--an empty element has no children or content-->
  <emptyElement2/> <!-- a shorthand notation for an empty element, used in this document-->
</configuration>
```

Note Distribution package file name:
The only restriction on the naming of your distribution package is it must have the .xml file extension.

Step 2 Pass the generated package distribution .xml file through the SSC postprocess command line utility, sscConfigProcess.exe. The sscConfigProcess utility performs the following required operations:

- Validates the preprocessed distribution package for both schema and business rule violations.
- Encrypts all credentials and secrets from their original clear text.
- Retrieves and packages any optional files referred to in the input file.
- Digitally signs the distribution package file to help prevent any tampering with its contents while it resides in the end station.

See “Postprocessing Utility” for a description of this utility.

Postprocessing Utility

The syntax of the postprocessing utility is:

sscConfigProcess [*input* | **-** | **-h**] [**-o** *output*]

Table 1-1 sscConfigProcess Command Elements

Command Elements	Meaning
input	Path, absolute or relative, to the distribution package xml file to be processed.
-	Utility reads file name from standard input.
-h	Displays utility version and command usage information.
(no option)	
-o *output*	Path, absolute or relative, to the processed distribution package xml file ready for deployment.
	If omitted, the output is sent to standard output.

Errors sent to the standard error output (stderr) include:

- usage errors (incorrect command)
- file I/O errors
- XML schema validation errors
- XML encryption errors
- XML signing errors
- Business rule violations

See Appendix C, "Postprocessing Verification Errors" for an overview of errors produced during postprocessing.

Note The sscConfigProcess utility requires that the following support files be located in the same folder:

- distributionPackage.xsd, schema file
- validateRules.xsl, business rules file

These files are provided in the SSCAdminUtils_{version}.zip file.

Note Both sscConfigProcess and sscPackageGen (discussed below) utilities use the Microsoft MSVCP71.dll file. This file is normally loaded into the system area when installing SSC. To allow for the use of these deployment tools in a non-SSC machine, this file is supplied in the SSCAdminUtils_{version}.zip file and should be located in the same folder as the utilities.

Distribution Package Deployment

Cisco assumes that the IT Administrators already have a preferred method of moving files to end-user stations (for example, Microsoft's SMS method).

Cisco provides a separate command line utility, sscPackageGen.exe, to facilitate the following enterprise deployment operations:

- Windows Installer single-step installation of a pre-configured SSC
- Windows Installer update of an initially deployed and installed SSC

Enterprise Deployment Utility

The syntax of the enterprise deployment utility is:

sscPackageGen {**insert** | **patch**} *source dest file*

Table 1-2 sscPackageGen Command Elements

Command Elements	Meaning
insert	Command to create a msi file.
patch	Command to create a msp file.
source	The full, absolute path for the input msi file.
dest	The full, absolute path for the output msi or msp file.
file	The full, absolute path for the input distribution package xml file.

Note The Cisco distributed (out-of-the-box) SSC installation file has the following generalized format:

Cisco_SSC-<os version>-<version>.msi

In particular, this translates to Cisco_SSC-XP2K-4_1_0_xxxx for the Windows XP/2000 version of the Cisco Secure Services Client.

The sscPackageGen utility uses PatchWiz.dll and mspatchc.dll files. These files are loaded at run-time. As a consequence, sscPackageGen.exe will run even if these dll files are not present. These files are required to create patches (the **patch** command), but are not required to configure an original package (the **insert** command). These two Microsoft files are part of the Windows Software Development Kit (SDK). These files may not be redistributed but can be freely obtained from the Microsoft web site as follows:

1. To search for the latest version, go to www.microsoft.com/downloads/.
2. From the Download Center window, choose **Developer Tools** in the Browse for Downloads list.
3. From the Developer Tools window, choose **Platform SDK** from the Show downloads for: drop-down list. Click **Go**.
4. From the Platform SDK window search for and choose the latest **Microsoft Windows Server 2003 Platform SDK Web Install**.
   At the time of the writing of this document, this was: Windows Server 2003 R2 Platform SDK Web Install. A direct link to this download is:

http://www.microsoft.com/downloads/details.aspx?FamilyID=0baf2b35-c656-4969-ace8-e4c0c0716adb&DisplayLang=en

5. Download and install the PSDK-x86.exe version of the Windows Server 2003 Platform SDK Web Install.
6. On the Installation Type window, choose **Custom installation**.
7. On the Custom Installation window, choose **Will not be available** for all features but **Microsoft Windows Installer SDK**.
8. Once installed, obtain the dll files from the following default install location:

   C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Samples\SysMgmt\Msi\Patching
9. Copy the dll files to the folder containing the sscPackageGen.exe utility.

Note The sscPackageGen utility checks the version of these two dll files before loading them. Only the following versions are acceptable. An error message is displayed when attempting to run the utility if this version check fails.

- PatchWiz.dll must be major version 3.
- mspatchc.dll must be major version 5.

End-User Initial Installation

Choose one of the following methods to initially install an end-user SSC.

- Enterprise deployment installation method
- Legacy installation method

Enterprise Deployment Installation Method (recommended)

SSC and its companion distribution package are deployed as a single file and installed in a single operation. (Recall that any required support files, optional CA certificates and optional FAST PACs, have already been added to the distribution package itself.) The sscPackageGen utility takes as input the out-of-the-box installation file (.msi) and the distribution package file (.xml) and creates a new pre-configured installation file (.msi).

Example 1-1 Initial Installation File

Create a pre-configured installation file, called *yourSSCInstallPkg.msi*, from the installation file obtained from Cisco (Cisco_SSC-XP2K-4_1_0_xxxx) and your validated and postprocessed distribution package file (distributionPackage.xml).

```
sscPackageGen insert C:\Cisco_SSC-XP2K-4_1_0_xxxx.msi C:\yourSSCInstallPkg.msi
C:\distributionPackage.xml
```

Deploying and executing yourSSCInstallPkg.msi on the end station will install SSC with your predefined distribution package configuration.

SSC supports a single-step, silent install by the standard Microsoft Installer mechanism. For this example, execute

msiexec /i yourSSCInstallPkg.msi /quiet /norestart.

Legacy Installation Method

A multistep operation (similar to releases earlier than Release 4.1) can also be used.

1. Deploy and install the installation file obtained from Cisco (Cisco_SSC-XP2K-4_1_0_xxxx).
2. Update the end-user configuration as outlined in the next section.

Updating End-User Configurations

Choose one of the following methods to update an end-user configuration.

- Enterprise deployment update method
- Legacy update method

Enterprise Deployment Update Method (recommended)

To update an initially deployed and installed SSC, the sscPackageGen utility takes as input either the original SSC installation file (.msi) or your pre-configured installation file (.msi) and the distribution package file (.xml) and creates a patch file (.msp).

Example 1-2 Update Based on Preconfigured File

Create an update patch file, called *yourSSCUpdatePkg.msp*, from your previously deployed preconfigured file and an updated distribution package file (.xml):

```
sscPackageGen patch C:\yourSSCInstallPkg.msi C:\yourSSCUpdatePkg.msp
C:\distributionPackage.xml
```

Note The updated distribution package file must have the same name as the original distribution package file, and the two must have different content.

Example 1-3 Update Bbased on Original Installation File

Create an update patch file, called *yourSSCUpdatePkg.msp*, from the original installation file obtained from Cisco and an updated distribution package file (.xml):

```
sscPackageGen patch C:\Cisco_SSC-XP2K-4_1_0_xxxx.msi C:\yourSSCUpdatePkg.msp
C:\distributionPackage.xml
```

Legacy Update Method

The deployment of a postprocessed distribution package .xml file (similar to releases earlier than Release 4.1) can also be performed.

Deploy the new/updated postprocessed distribution package .xml file into the following folder created by the SSC installer:

<install folder>\distribution, where the default <install folder> is: Program Files\Cisco Systems\Cisco Secure Services Client.

Upgrading End-User Installations

There are two scenarios for updating an end-user installation with Release 4.1:

- updating Release 4.1 with a newer maintenance release
- updating Release 4.0 (any maintenance release)

Release 4.1

Upgrading the Cisco SSC Release 4.1.x to a later release is the same process as the initial installation described above.

All previously deployed (locked) networks will be replaced by those in the updated distribution package file. Therefore when adding new networks or modifying an existing network you must also include in the updated configuration file any unaltered network that you want to keep. Deleting any previously existing network in the updated distribution package will delete that network.

Release 4.0

Upgrading an earlier Cisco SSC Release 4.0.x to Release 4.1.x is the same process as the initial installation described above.

When you update an administrator version (all networks are user-defined), the user-configured networks are migrated to the upgraded version. However if there is a user created network for a wired network or for one of your enterprise SSIDs, and the distribution package also configures one or more of these networks, then the original network profiles are replaced with the new distribution package (locked) versions.

When you update a deployed end-user version, all existing administrator deployed (locked) networks are replaced by the set of new (locked) networks in the distribution package. The user-configured networks are migrated to the upgraded version. However if there is a user created network for a wired network or for one of your enterprise SSIDs, and the distribution package also configures one or more of these networks, then the original network profiles are replaced with the new distribution package (locked) versions.

Pre-Installation of Client Certificates

If the end-user SSC is using a client-certificate-based EAP method, then the client certificate used to supply the user's credentials must be independently deployed and placed in the proper Windows Certificate Store (User-Personal Store). The distribution package file does not include deploying client certificates.