CHAPTER 1

Enterprise Deployment

This chapter contains the following sections:

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 configuration file does the IT Administrator have full access to all settings and network configurations.

- Default download package—contains a default configuration that is configured with a non-expiring, wired only license. You can download a trial wireless license from cisco.com at this URL:

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

When 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 configuration 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.

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

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

The trial license adds support for:

- Wireless (802.11) network adapters
- Additional EAP methods: LEAP, EAP-PEAP, EAP-TTLS, EAP-MD5

Supported Operating System Environments

The supported operating system environments are:

- Windows XP Professional (SP1, SP2), Windows 2000 (SP4), or Windows 2003 server

Note 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 consists of the configuration file which contains the following functional blocks:

- License

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

- Policy
  - User control policy

    Sets the 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.

- Groups

  A group, fundamentally, is a collection of configured connections (networks). Every configured connection must belong to some group or be defined under the *globalNetworks* section in the distribution package.

  Note End-users can add networks only to groups and not to the *globalNetworks* section (because they typically do not have access to the management tool that would allow them to sign the distribution package).

  Classifying connections into groups provides multiple benefits:

  – Improved user-experience when attempting to make a connection. It is important to understand how the client establishes a network connection in order to illustrate this point. The client works through the list of available networks in the order in which they are defined until a successful connection is made.

    For example, an enterprise end-user who travels often outside the business campus might configure connections for public WiFi networks or hotspots. Without groups, a newly configured home network is added to the end of this list, which could be quite large. The client works through the list from the beginning, including all the public networks, before establishing a connection to the home network. This greatly increases the time to get connected to the last added network.

  – Easier management of configured connections. In the previous example, if an end-user attempts to delete some connections to get connected quicker, the deleted connections might be needed at a later time. However, if the connection list is divided into groups, each list would be much smaller. When using groups, it is easy to switch between the groups to obtain faster connectivity.

  A group may be created by an administrator or an end-user. There must be at least one group defined in the configuration. If there are multiple groups, one group must be chosen as the *active* group and the client attempts to make a network connection using the connections defined in the active group. End-users can add or delete networks only from the active group. Groups can be added or deleted by clicking on the *Configure Groups* button on the main screen of the client GUI.

  Networks that are defined in the *globalNetworks* section of the distribution package are available in every group at the top of the list. Because only enterprise administrators can create *globalNetworks*, this provides an administrator with control over the enterprise networks that an end-user can connect to, even in the presence of user-defined networks. An end-user is not able to delete administrator configured networks.

  It is important to note that a typical end-user of an enterprise network does not need to have a knowledge of groups in order to use this client. It is the responsibility of the administrator to always specify a default group in the created distribution package. If there is just one group available, the client selects that as the active group. The end-user can add or delete their own networks without using groups.

  Note A group selection is not maintained across reboots or repairs of the client. When the client is repaired or restarted, the client always goes back to the first configured group in the configuration.xml file.

- Networks

  Networks contain 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" 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, SSCMgmtToolkit_{release}.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

Click **Download Software > Client Adapters and Client Software** and follow the prompts 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, configuration.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.

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

Cisco supports two basic methods for creating your distribution package xml instance file:

- Methods Based on the Language of the Schema—the manual process supported in releases earlier than Release 4.2
- Methods Based on Descriptive English—a wizard utility

Methods Based on the Language of the Schema

Follow these steps to create a distribution package file.

Step 1 Generate the descriptive .xml distribution package file as specified by the SSC schema. 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 to create an XML instance file either from scratch or by cut-and-paste from known examples.

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 serves 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.

- Illegal XML characters must be replaced by the following entity references. Entity references always start with the '&' character and end with the ';' character.

  less than—use < for the character <
  greater than— use > for the character >
  ampersand—use & for the character &
  apostrophe—use ' for the character '
  quotation mark—use " for the character "

- 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 name of your distribution package must be configuration.xml.

Step 2 Pass the generated package distribution .xml file through the SSC postprocess command line utility, sscManagementUtility.exe. The sscManagementUtility 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 (the distribution .xml file that was just generated). The optional files include the PACs and the CA certificates.
- 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 command-line description of this utility.

Methods Based on Descriptive English

Cisco provides a wizard that walks you through the distribution package file creation process. The GUI version of the sscManagementUtility allows you to:

- Create a validated and signed distribution package from scratch
- Import an existing unsigned file to use as a starting point for making changes
- Postprocess an existing distribution package

The GUI version of the sscManagementUtility supports creating and processing distribution package xml files for all versions of SSC Release 4.1 and later.

Execute **sscManagementUtility** to open the utility. Invoking the utility, starts the GUI

Postprocessing Utility

The syntax of the command-line version of the postprocessing utility is shown below. .

sscManagementUtility.com {**help** | **validate** | **sign**} [command specific arguments]

sscManagementUtility.com help

sscManagementUtility.com validate {**-i** *input-file* | **--in=***input-file*}

sscManagementUtility.com sign {**-i** *input-file* | **--in=***input-file*} {**-o** *output-file* | **--out=***output-file*}

Table 1-1 sscManagementUtility Command Elements

Command Elements	Meaning
validate	Validate a distribution package xml file only.
sign	Postprocess (validate, encrypt, sign) a distribution package xml file.
help	Displays utility release and command usage information.
-i *input-file* **--in=***input-file*	Path, absolute or relative, to the distribution package xml file to be processed.
-o *output-file* **--out=***output-file*	Path, absolute or relative, to the processed distribution package xml file ready for deployment.

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

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

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

Note The utility (sscManagementUtility.com) requires the following support files. These files are provided in the SSCAdminUtils_{release}.zip file in a data folder that is structured by SSC version. This folder structure must be left intact when extracting the contents of the zip file.

- configuration.xsd, schema file

  Release numbering is defined in the schema itself. Each instantiated distribution package xml file retains the release numbering scheme of its associated schema file.

- validateRules.xsl, business rules file

  Release numbering is controled by a namespace for the file, as follows:

  xmlns:validateRules="http://www.cisco.com/2007/CSSCValidationRules/A.B.C", where A, B and C correspond to major, minor and maintanence, respectively.

Note The management utility uses the Microsoft msvcp71.dll and msvcr71.dll files. These files are normally loaded into the system area when installing SSC. To allow for the use of these deployment tools in a non-SSC machine, these files are supplied in the SSCAdminUtils_{release}.zip file and should be left in the same folder as the utility.

Additionally, the GUI version of the utility uses several supplied QT dll files. These should also be left in the same folder as the utility.

Distribution Package - SSC Release Compatibility

Release Numbering for SSC

The management toolkit package (.zip) file and previous releases of the installation file (.msi) obtained from Cisco have the following format:

SSCMgmtToolkit_A.B.C.xxxx.zip or Cisco_SSC-{OS}-A_B_C_xxxx.msi

For the Windows 2000/XP release of SSC, this becomes:

SSCMgmtToolkit_A.B.C.xxxx.zip or Cisco_SSC-XP2K-A.msi, where *A* indicates major release change.

Compatibility Between SSCMgmtToolkit and SSC

The following table lists the release of the management utility package that may be used to produce a full-featured distribution package for the designated release of SSC.

Table 1-2 Management Utility vs. SSC

This Release of Management Toolkit Package	Supports These SSC Releases
SSCMgmtToolkit_5.0.0.xxxx.zip	Cisco_SSC-XP2K-4_1_0_xxxx.msi
	Cisco_SSC-XP2K-4_1_1_xxxx.msi
	Cisco_SSC-XP2K-4_1_2_xxxx.msi
	Cisco_SSC-XP2K-4_2_0_xxxx.msi
	Cisco_SSC-XP2K-5.msi

Compatibility Between Distribution Package and SSC

SSC Release 5.0 is a major software release and employs a new schema. This schema is not compatible with the schema of prior SSC releases. To aid in the translation of the old schema to the new schema, a schema conversion tool is provided. For additional information see the "Upgrading SSC Release 4.1.x Installations to SSC Release 5.0" section on page 1-12.

This conversion tool will not convert an administrator created SSC Release 4.1 distribution package (schema version 4.1.x) to the SSC Release 5.0 schema. Instead, it will use SSC Release 4.1 internal configuration files (files in *Program Files\Cisco Systems\Cisco Secure Services Client*) to translate the administrator configured networks to the SSC Release 5.0 schema.

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

Note Deployment by means of remote desktop is not supported.

Enterprise Deployment Utility

The enterprise deployment utility (sscPackageGen) 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). The syntax of the utility is:

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

Table 1-3 sscPackageGen Command Elements

Command Elements	Meaning
insert	Command to create a msi 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.

End-User Initial Installation

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

- Enterprise deployment installation method
- Legacy installation method (recommended)

Enterprise Deployment Installation Method

SSC and its companion distribution package are deployed as a single file and installed in a single operation. Recall that any required support files (CA certificates and PACs) have already been added to the distribution package itself.

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-5) and your validated and postprocessed distribution package file (configuration.xml).

```
sscPackageGen insert C:\Cisco_SSC-XP2K-5.msi C:\yourSSCInstallPkg.msi
C:\configuration.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.

(The parameter *norestart* prevents a silent install from rebooting the PC.)

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-5).
2. Update the end-user configuration as outlined in the next section.

Note SSC Release 5.0 and later uses an intermediate driver to control the network adapters. Installation is stopped and the user is informed if it detects the presence of another driver with which SSC is not able to co-exist. You need to either disable or un-install the conflicting application.

Updating End-User Configurations

The legacy update method is used to update an end-user configuration.

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

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

   C:\Documents and Settings\All Users\Application Data\Cisco\
   Cisco Secure Services Client\newConfigFiles

2. Either restart the Cisco Secure Services Client service or from the Help menu, choose **Repair**.

Note SSC also detects and implements the new configuration file whenever it attempts a new connection.

Upgrading SSC Release 4.1.x Installations to SSC Release 5.0

There are two components to upgrading existing SSC 4.1.x releases to SSC Release 5.0:

- All previously deployed administrator (locked) networks from SSC Release 4.1.x must be upgraded to SSC Release 5.0.
- All end-user created networks from SSC Release 4.1.x must be upgraded to SSC Release 5.0

Upgrading Administrator Deployed Networks from SSC Release 4.1.x to SSC Release 5.0

An administrator must have the following SSC Release 5.0 client elements on his PC:

- SSC Release 5.0 installation msi file (Cisco_SSC-XP2K-5.msi)
- Configuration management utility (SSCMgmtToolkit_5.0.0.xxxx.zip)
- Configuration combining tool (ConfigCombiner.exe)
- Configuration conversion tool (ConfigConverter.exe)
- Administrator xslt file (configConvert_3_1_admin.xslt)—used to translate administrator-configured SSC Release 4.1 networks to SSC Release 5.0 schema.
- sscPackageGen that generates a custom installation package

The administrator also must have the current SSC Release 4.x deployment package, translated into SSC Release 4.1.2 internal configuration. This is the *profiles* folder found under the *Program Files\Cisco Systems\Cisco Secure Services Client* folder.

In order to deploy an SSC Release 5.0 client that is equivalently configured to your SSC Release 4.x distribution, you must perform these operations:

1. Use the combining tool (ConfigCombiner.exe) to combine SSC Release 4.1 configuration files into a single file:

   Usage: ConfigCombiner.exe [options]

   Options include:

   --source *directory* or -s *directory*—specifies the source directory path. If the source directory option is not specified, the default value for the source directory is *C:\Program Files\Cisco Systems\Cisco Secure Services Client\profiles*.

   --quiet or -q—do not dump the result

   --help—gives the usage of the tool

   The following illustrates a conbining tool example:

   ```
   ConfigCombiner.exe -q
   ```

   The output of this operation produces a file called *configuration.xml*. The file is located in the folder where the tool was executed. The file contains the information in the multiple folders under *c:\Program Files\Cisco Systems\Cisco Secure Client Services\profiles*.

   Note SSC Release 4.1.x files are not modified in any way as a result of this operation.

2. Use the conversion tool (ConfigConverter.exe) with the administrator XSLT file (configConvert_3_1_admin.xslt) to convert the output of the combining tool into an SSC Release 5.0 configuration.xml file:

   Usage: ConfigConverter.exe [options]

   Options include these values:

   --quiet or -q—specifies do not dump the result

   --output *filename* or -o *filename*—specifies the output XML file

   --input *filename* or -i *filename*—specifies the input XML file

   --xslt *filename* or -xslt *filename*—specifies the XSLT file

   You should specify the *--xslt* file option with the XSLT file name set to **configConvert_3_1_admin.xslt** when you are converting the administrator deployed networks using the ConfigConverter tool. This is the same tool used with a different default xslt file to translate the end-user created networks on end-user systems.

   The following illustrates a conversion tool example:

   ```
   ConfigConverter.exe -i configuration.xml -o configuration.xml
   --xslt configConvert_3_1_admin.xslt
   ```

   The output of this operation is a SSC Release 5.0 schema compatible distribution package with an equivalent configuration of your SSC Release 4.1.x deployed networks.

3. You can now use the management utility to perform these operations:
   - Read in the SSC Release 5.0 configuration.xml (which contains the administrator deployed SSC Release 4.1 networks)
   - If needed, modify the SSC Release 5.0 configuration.xml file and root
   - Sign the SSC Release 5.0 configuration.xml file
4. Run the packageGen tool to bundle the signed configuration.xml file along with the SSC Release 5.0 msi file and then deploy the package.

Upgrading End-User Created SSC Release 4.1.x networks to SSC Release 5.0

When SSC Release 5.0 is installed on a PC as an upgrade, it automatically upgrades the SSC Release 4.1.x end-user created networks to SSC Release 5.0 networks. There is nothing that you, the administrator, or the end-user need to do. The results of the upgrade is as follows:

- SSC Release 5.0 starts running with the deployed administrator configuration file.
- All the end-user created profiles from SSC Release 4.1 are imported into the SSC Release 5.0 client.
- This conversion is done once only during the upgrade.
- SSC Release 4.1 has multiple user xml files on an end-station, but SSC Release 5.0 has only one user-XML file. The conversion tool places the contents of multiple SSC Release 4.1 user-profile files into the single SSC Release 5.0 user XML file. Each user XML file in SSC Release 4.1 corresponds to a group in SSC Release 5.0. The group name is the user xml file name prefixed with *CSSC4_*. The profiles in the *allusers* file is placed in the *CSSC4_allusers* group. It is the responsibility of the end-user to later go through the list of available networks using the GUI and delete any networks they do not want.
- There may be multiple networks created in SSC Release 5.0 for a single network in SSC Release 4.1. This is because the SSC Release 5.0 schema allows only one EAP-method per network, whereas the SSC Release 4.1 schema allows multiple EAP methods per network. This means that a user network from SSC Release 4.1, after conversion to SSC Release 5.0, has a network name that includes both the SSC Release 4.1 network name and the EAP method. This is done to help avoid confusion.
- On an upgrade from SSC Release 4.1 to SSC Release 5.0, all static user credentials are imported into SSC Release 5.0. Also theWEP and PSK credentials input by the user are also imported into SSC Release 5.0. However, any 802.1x credentials are not imported, they need to be re-entered if required.

Pre-Installation of Client Certificates

If the end-user SSC file uses 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 deploy client certificates.