CHAPTER 1

About the SOAP LEG

This module describes the SOAP LEG software module, and terms and concepts.

Information About the SOAP LEG

The SCMS SM SOAP LEG is a software module that can query an external server via the Simple Object Access Protocol (SOAP) in order to obtain additional information for the subscribers that were logged-in to the SM via various APIs and LEGs. The main purpose of the SOAP LEG is to define the policy of the subscriber based on the input data, the package association configuration, and the query results.

The LEG can query any external server via the SOAP communication protocol if the external server implements an interface defined by the SCMS SM SOAP LEG.

The SCMS SM SOAP LEG supports SOAP 1.1.

The SCMS SM SOAP LEG is an extension of the Subscriber Manager (SM) software and runs as part of the SM.

- SOAP Integration Overview, page 1-1
- Common Topologies, page 1-3
- Terms and Concepts, page 1-5

SOAP Integration Overview

The SM activates the SOAP LEG in order to obtain the policy value (or part of the policy value) for the subscribers that are already logged in to the SM.

With the data that the SOAP LEG receives from the SM, it creates a SOAP request, which it issues to the external server in order to retrieve the policy value. After the external server replies, the SOAP LEG determines the policy value according to the input data, the package association configuration, and the query results. It then initiates a subscriber login to the SM. For more information about the package association, see Information about Configuring the Package Association, page 3-3.

- Query Interface, page 1-2
- Secure Requests, page 1-2
- Implementing Query Interface at the Server, page 1-2

Query Interface

The SOAP installation package includes a WSDL file. This WSDL file defines the SOAP LEG query to the external server:

```
QuerySubscriberOut querySubscriber(QuerySubscriberIn subIn)
```

The **QuerySubsriberIn** parameter contains the following data:

- **subscriberId** —Contains the ID of the subscriber
- **mappings** —Contains the Network IDs of the subscriber
- **keys/values** —May contain additional data that the external server may need in order to perform the query

The Web Server responds to the query and SOAP LEG analyzes the results. The output of the Web Server (QuerySubscriberOut) consists of the following elements:

- **subscriberId** —Contains the ID of the subscriber
- **mappings** —Contains the Network IDs of the subscriber
- **keys/values** —May contain additional data that the SOAP LEG may need in order to determine the package value
- **propertyKeys/propertyValues** —May contain subscriber properties; for example, packageId or monitor.

Note that **keys** and **values** are used internally by the LEG for the package association procedure and are not passed to the SM when the subscriber is logged in.

Upon receiving a reply from the Web Server, the SOAP LEG adds the query output values to the query input values. Following this, if the SOAP LEG is configured to do so, the LEG uses this data as the input for the package association procedure. See Information about Configuring the Package Association, page 3-3.

Secure Requests

The SOAP LEG is able to issue a secure request to the external server using the UsernameToken profile as defined in the WS-Security specification. Specifically, it attaches username and password to every SOAP request it sends. For further information on configuring the username and password, see Using the SOAP LEG Command-Line Utility (CLU), page 4-1.

Note The SOAP LEG supports only text passwords.

Implementing Query Interface at the Server

Integrating the external server with the SOAP LEG is a two stage process:

1. Compile the provided WSDL file using one of the various tools available. For example, Apache Axis can be used (http://ws.apache.org/axis/). The WSDL file is included in the Cisco WSDLmodule.
2. Provide the implementation of the **querySubscriber** function according to the server business logic.

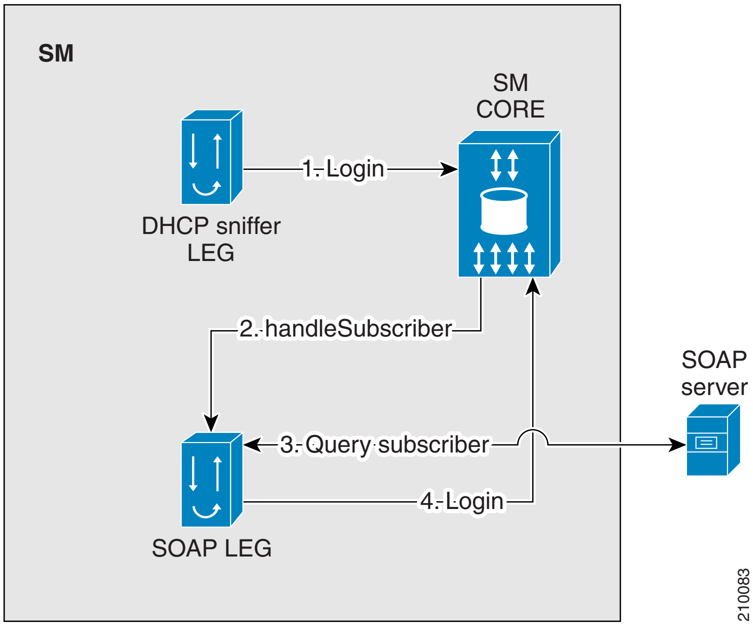
Common Topologies

You can use the SOAP LEG in any SM topology, providing it is possible to supply the LEG with the information it needs in order to perform the query to the policy server and determine the subscriber policy.

The following figures show the most common topologies.

The following figure shows the topology with the SM API:

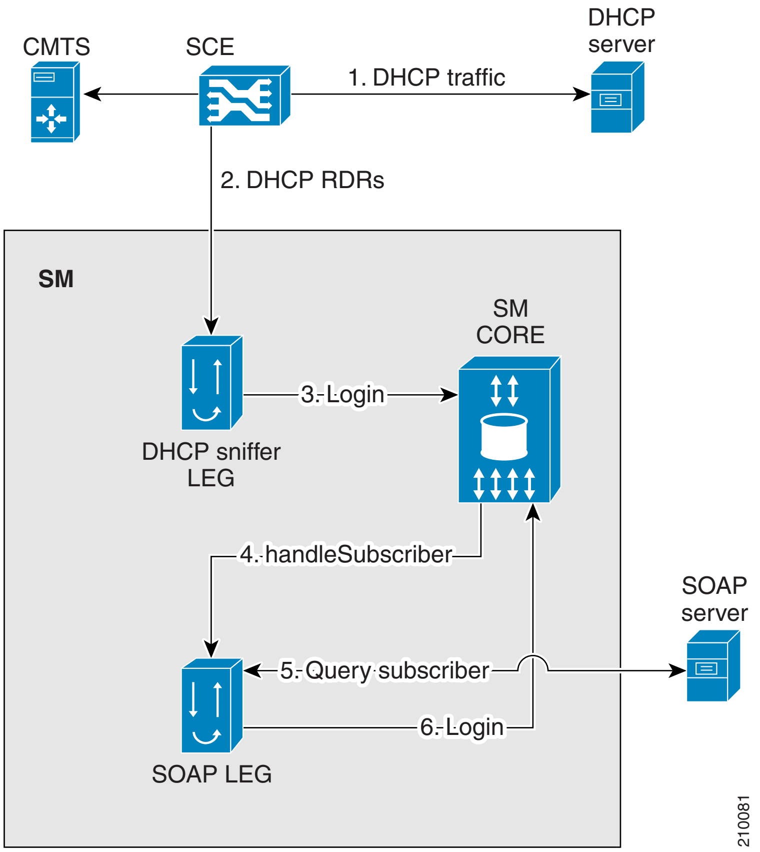
Figure 1-1 SOAP Topology with SM API



The SM API performs a login operation to the SM (1). The SM identifies that theSOAP LEG needs be activated, and therefore it does not perform a subscriber login at this stage. The SM core passes the information received from the SM API to the SOAP LEG (2). The SOAP LEG queries the SOAP server and identifies the relevant packageId based on the configuration, input parameters, and the query results (3). The SOAP LEG then performs a login operation to the SM (4).

The following figure shows the topology with the DHCP Sniffer LEG:

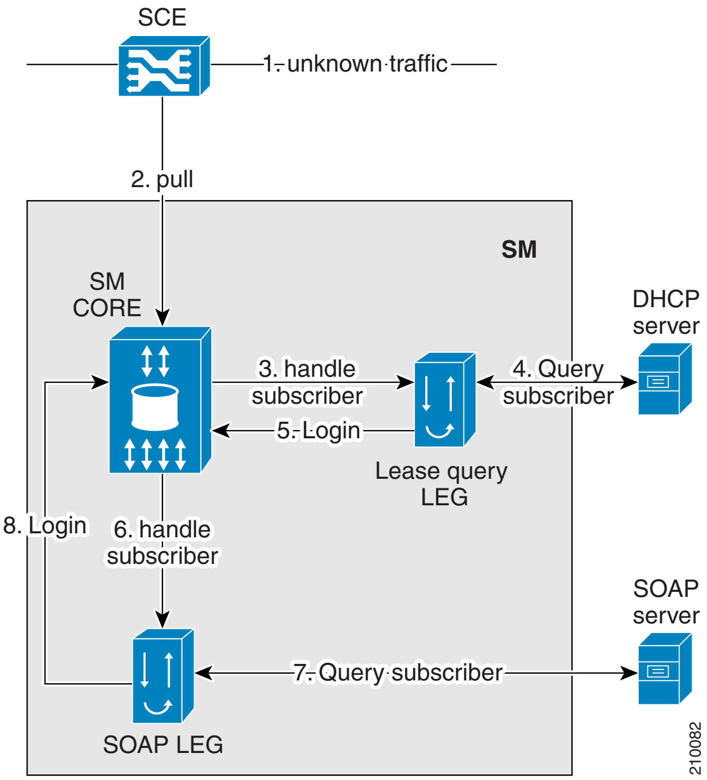
Figure 1-2 SOAP LEG Topology with DHCP Sniffer LEG



The DHCP traffic passes through the SCE (1), which sends a DHCP RDR to the DHCP Sniffer LEG (2). The DHCP Sniffer LEG extracts the relevant information and performs a login operation to the SM (3). The SM identifies that the SOAP LEG needs to be activated, and therefore it does not perform a subscriber login operation at this stage. The SM core passes the information received from the DHCP Sniffer LEG to the SOAP LEG (4). The SOAP LEG queries the SOAP server and identifies the relevant packageId based on all the information received and the query results (5). The SOAP LEG then performs a login operation to the SM (6).

The following figure shows the topology with the DHCP Lease Query LEG:

Figure 1-3 SOAP LEG Topology with DHCP Lease Query LEG



Unknown traffic passes through the SCE (1), which issues a pull request to the SM (2). The SM issues an anonymous-pull-request to the DHCP Lease Query LEG (3). The DHCP Lease Query LEG then queries the DHCP server (4), after which it performs a login operation to the SM (5). The SM identifies that the SOAP LEG needs to be activated, and therefore it does not perform a subscriber login at this stage. The SM Core passes all of the information received from the DHCP Lease Query LEG to the SOAP LEG (6). The SOAP LEG queries the SOAP server and identifies the relevant packageId based on the information received and the query results (7). The SOAP LEG then performs a login operation to the SM (8).

Terms and Concepts

The following terms and concepts are necessary to understand the SOAP LEG and SM configuration and operation. Additional information can be found in the *Cisco SCMS Subscriber Manager User Guide.*

- LEG (Login Event Generator), page 1-6
- Subscriber ID, page 1-6
- Subscriber Network IDs (mappings), page 1-6
- Subscriber Package, page 1-6
- SOAP, page 1-6
- WSDL, page 1-6
- WSS, page 1-6
- UsernameToken Profile, page 1-7

LEG (Login Event Generator)

A software component that performs subscriber login and logout operations on the SM, which is used to handle dynamic subscriber integration.

Subscriber ID

The Service Control solution requires a unique identifier for each subscriber. A subscriber ID represents a logical subscriber entity from the service provider perspective.

Subscriber Network IDs (mappings)

The subscriber network IDs (mappings) are a list of network identifiers, such as IP addresses or VLANs. The SCE uses these identifiers to associate network traffic with subscriber records.

Subscriber Package

A subscriber policy package usually defines the policy enforced by Cisco SCMS solutions on each subscriber. The main function of the SOAP LEG is to determine a package value based on the input parameters, configuration, and query results. For additional information, see the *Cisco Service Control Application for Broadband User Guide.*

SOAP

SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.

WSDL

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services).

WSS

WS-Security (Web Services Security) is a communications protocol providing a means for applying security to Web Services. Originally developed by IBM, Microsoft, and VeriSign, the protocol is now officially called WSS and is developed and maintained via committee in Oasis-Open.

The protocol contains specifications on how integrity and confidentiality can be enforced on Web Services messaging. WS-Security incorporates security features in the header of a SOAP message and thus works in the application layer. Thus, it ensures end-to-end security.

UsernameToken Profile

The <wsse:UsernameToken>is an element introduced in the WSS SOAP Message Security documents as a way of providing a username.