APPENDIX A

Security

This appendix describes the security mechanisms used in a Subscriber Edge Services Manager (SESM) application.

The Cisco SESM:

- Is built using Java technology based on the J2EE specification. As such, it inherits the security features both of the Java language platform and the security framework in J2EE.
- Is a web server-based application, and so must be deployed in a web server that enforces HTTP security.
- Plays a role in authentication for the user, so it must also enforce constraints at this level.

Java Platform Security Description

The following URLs provide a description of Java platform security:

- http://java.sun.com/security/index.html
- For specific Java platforms:
  - http://java.sun.com/products/jdk/1.2/docs/guide/security/
  - http://java.sun.com/products/jdk/1.3/docs/guide/security/
- For training:
  - http://developer.java.sun.com/developer/onlineTraining/Security/Fundamentals/index.html
- For miscellaneous articles:
  - http://developer.java.sun.com/developer/technicalArticles/Security/

HTTP Security Description

HTTP security involves two separate issues:

- Encryption of communications using HTTPS
- Basic and digest access authentication in HTTP1.1 (RFC 2617)

HTTPS Description

HTTPS (Secure Hypertext Transfer Protocol) is HTTP over Secure Sockets Layer (SSL), which are HTTP packets sent as encrypted data. This is the mechanism by which data is securely transmitted over the Internet between a browser client and a server.

SESM implements SSL using the Java Secure Sockets Extension (JSSE). For information about JSSE, go to:

http://java.sun.com/products/jsse/

The J2EE specifications describe an extension framework for the integration of SSL implementations. For implementations other than JSSE, go to:

http://www.phaos.com/e_security/prod_ssl.html

Keytool and Keystore

The SSL part of HTTPS requires a certificate to generate the encryption key. For the Jetty web server bundled with the Cisco SESM, the certificate is named keystore and is found in the /etc directory. The keystore file is created by the keytool utility. For detailed instructions on the use of keytool, go to the following URL:

http://java.sun.com/products//jdk/1.2/docs/guide/security/SecurityToolsSummary.html

The sample keystore functions for nonproduction deployments. However, you must obtain a site-specific certificate for production deployments from VeriSign, Inc. at:

http://www.verisign.com

Though certificates are generally the same in concept, they tend to differ in implementation. Therefore, a degree of certificate manipulation is required to obtain a certificate from a given source to work with a given SSL implementation. For JSSE and the Jetty web server, the required steps are described at:

ftp://jetty.mortbay.com/pub/Jetty-dev/webapps/jetty/JsseSSL.html

For other implementations, go to:

http://www.openssl.org

The keystore file is a certificate used for secure sockets layer (SSL) encryption. The SSL implementation shipped with the Cisco SESM is of commercial quality and can use certificates generated by keytool. Keytool resides in the same directory as the JRE.

Caution A keystore is required for deployments that use HTTPS. HTTPS does not function without a valid keystore file. The file included with the installation works, but you should replace it with a keystore valid for your specific deployment.