SESM Security

This appendix describes the security mechanisms used in a Subscriber Edge Services Manager (SESM) deployment. This chapter includes the following topics:

Java Platform Security References

SESM applications inherit the security features of the Java language platform and of the J2EE framework. The following URLs describe security topics related to the Java and J2EE technology:

- For Java security software and documentation:

  http://java.sun.com/security/index.html
- For information related to JDK 1.3:

  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/

Using HTTPS in SESM Portals

This section contains the following topics concerning HTTPS:

HTTPS References

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 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/1.3/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:

http://jetty.mortbay.com/jetty/doc/SslListener.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 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.

Configuring SESM Portals to Run on SSL Ports Only

The sample applications installed with SESM provide an option on the logon page that allows the subscriber to choose between starting a secure (HTTPS) session or a standard (HTTP) session. The default configuration files start both types of listeners: one HTTP listener and one HTTPS listener to support either choice from the logon page.

To remove this option from the logon page and run the portal in secure mode only, follow these procedures:

Step 1 To remove the secure or standard session option from the NWSP logon page, comment out the HTML in accountLogonBody.jsp.

```
<%-- Make this page either secure or insecure --%>
<% if (request.isSecure()) { %>
<tr>
<td colspan=2 align=center class="MediumText">
<A HREF="/insecure/home">
<l10n:resource key="standardLoginLabel">Standard</l10n:resource>
</A>
  |  
<l10n:resource key="secureLoginLabel">Secure</l10n:resource>
</td>
</tr>
<% } else {%>
<tr>
<td colspan=2 align=center class="MediumText">
<l10n:resource key="standardLoginLabel">Standard</l10n:resource>
  |  
<A HREF="/secure/home">
<l10n:resource key="secureLoginLabel">Secure</l10n:resource>
</A>
</td>
</tr>
<% } %>
```

Step 2 In the Jetty configuration file, comment out or remove the call that starts the standard HTTP listener.

For example, in nwsp.jetty.xml, surround the addListener call for the http.socketListener class with comment indicators, as shown here:

```
<!-- (start comment)
<Call name="addListener">
<Arg>
<New class="org.mortbay.http.SocketListener">
<Set name="port"><SystemProperty name="application.portno" default="8080"/></Set>
<Set name="minThreads">5</Set>
<Set name="maxThreads">255</Set>
<Set name="maxIdleTimeMs">60000</Set>
<Set name="maxReadTimeMs">60000</Set>
</New>
</Arg>
</Call>
(end comment) -->
```

Step 3 In the generic startup script, remove the information that defines and opens a port for standard HTTP traffic.

The generic script is executed by all of the application-specific startup scripts. In start.sh or start.cmd, change:

```
MGMTPORTNO=`expr $PORTNO + 100`
SSLPORTNO=`expr $PORTNO - 80 + 443`
PORTS="$PORTNO $MGMTPORTNO $SSLPORTNO"
```

to:

```
MGMTPORTNO=`expr $PORTNO + 100`
SSLPORTNO=1234
PORTS="$MGMTPORTNO $SSLPORTNO"
```

Further down in the script, delete the -Dapplication.portno=$PORTNO argument, shown in bold below:

```
$JAVA -Xms64m -Xmx64m \
-classpath $CLASSPATH \
-Dinstall.root=$INSTALLDIR \
-Djetty.home=$JETTYDIR \
-Dapplication.home=$APPDIR \
-Dapplication.log=$LOGDIR \
-Dapplication.portno=$PORTNO \
-Dapplication.ssl.portno=$SSLPORTNO \
-Dmanagement.portno=$MGMTPORTNO \
$MODE \
$JVMOPTIONS \
com.cisco.sesm.jmx.Main \
$JETTYDIR/config/$APP.jetty.xml \
$DESSDIR/config/config.xml \
$LIBDIR/config/config.xml \
$APPCONFIGDIR/$APP.xml \
```

Step 4 If you are running a captive portal solution, change the configured redirections to the NWSP application to use the HTTPS protocol and the HTTPS port you defined in the generic startup script.

In the captiveportal.xml file, change the following lines. The port numbers must match the SSL port number defined in the serviceportal configuration (which in the default configuration is nwsp.xml.

```
<Set name="userRedirectURL">
http://<SystemProperty name="serviceportal.host" default="nwsp"/>:
<SystemProperty name="serviceportal.port" default="8080"/>/home</Set>
<Set name="serviceRedirectDefaultURL">http://nwsp:8080/serviceRedirect</Set>
<Set name="errorURL">
    http://<SystemProperty name="serviceportal.host" default="nwsp"/>: <SystemProperty
    name="serviceportal.port" default="8080"/>/home</Set>
```

to

```
<Set name="userRedirectURL">
    https://<SystemProperty name="serviceportal.host "default="nwsp"/>:
    <SystemProperty name="serviceportal.port" default="1234"/>/home</Set>
<Set name="serviceRedirectDefaultURL">https://nwsp:1234/serviceRedirect</Set>
<Set name="errorURL">
    https://<SystemProperty name="serviceportal.host" default="nwsp"/>: <SystemProperty
    name="serviceportal.port" default="1234"/>/home</Set>
```

Step 5 If you are using the Message Portal application in your captive portal solution, change the configured redirections to NWSP to use the HTTPS protocol and the HTTPS port you defined in the generic startup script.

In messageportal.xml, change the following lines:

```
<Set name="defaultURL">
    http://<SystemProperty name="serviceportal.host" default="nwsp"/>:
    <SystemProperty name="serviceportal.port" default="8080"/>/</Set>
```

to:

```
<Set name="defaultURL">
    https://<SystemProperty name="serviceportal.host" default="nwsp"/>:
    <SystemProperty name="serviceportal.port" default="1234"/>/</Set>
```