Unified Management Portal

Secure Sockets Layer Implementation Guide

6.2

Document Revision History

Document Version	Date	Changes
Beta	05/01/2012	Beta release.
1.0	08/01/2012	Initial release.
1.1	09/15/2012	Updated version number for UMP 6.0.
1.2	12/18/2012	Initial version for UMP 6.2. Updated order for importing root, intermediate, and entity certs issued by a certificate authority.

Legal Notices

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Nimsoft LLC disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Nimsoft LLC shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Nimsoft LLC and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Nimsoft LLC as governed by United States and international copyright laws.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Nimsoft LLC's standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Trademarks

Nimsoft is a trademark of CA.

Adobe®, Acrobat®, Acrobat Reader®, and Acrobat Exchange® are registered trademarks of Adobe Systems Incorporated.

Intel® and Pentium® are U.S. registered trademarks of Intel Corporation.

Java(TM) is a U.S. trademark of Sun Microsystems, Inc.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Netscape(TM) is a U.S. trademark of Netscape Communications Corporation.

Oracle® is a U.S. registered trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of the Open Group.

ITIL® is a Registered Trade Mark of the Office of Government Commerce in the United Kingdom and other countries.

All other trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

For information on licensed and public domain software, see the *Nimsoft Monitor Third-Party Licenses and Terms of Use* document at: http://docs.nimsoft.com/prodhelp/en_US/Library/index.htm?toc.htm?1981724.html.

Contact Nimsoft

For your convenience, Nimsoft provides a single site where you can access information about Nimsoft products.

At http://support.nimsoft.com/, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- Nimsoft Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about Nimsoft product documentation, you can send a message to support@nimsoft.com.

Contents

Chapter 1: Introduction

This document describes how to configure a Secure Sockets Layer (SSL) connection in order to access UMP via HTTPS. It provides instructions for setting up a self-signed certificate, or a certificate signed by a Certificate Authority (CA). In addition, this document includes instructions for implementing a wildcard certificate.

This document uses a scenario-based approach. This means that each chapter (after this initial chapter) provides self-contained instructions for a specific scenario. These scenarios are as follows:

- Chapter 2 - Implement a 1024-bit Self-Signed SSL Certificate (see page 11)
- Chapter 3 - Implement a 2048-bit Self-Signed SSL Certificate (see page 13)
- Chapter 4 - Implement a CA-Signed SSL Certificate (see page 19)
- Chapter 5 - Implement a Wildcard SSL Certificate (see page 27)

This chapter (Chapter 1) provides important information to know before you get started.

There are various factors that can affect the steps you use to implement SSL in your environment. Troubleshooting SSL Certificates (see page 31) provides help with some of the common issues you may encounter.

The wasp and SSL with UMP

The purpose of this section is to provide background information on the wasp as it relates to the activities described in this guide. The actual instructions for configuring the wasp are provided in the scenarios in each chapter.

Configuring UMP for SSL entails configuring the wasp. The wasp (Web Application Service Provider) is an embedded Tomcat web server running as a probe. It is distributed to the system during the UMP installation, and afterward, appears as a probe in Infrastructure Manager.

Regardless of the certificate you intend to implement, the first step in each scenario in this guide is to modify the wasp.cfg file to enable HTTPS. When this change takes effect, the following occurs:

- The wasp.keystore, an encrypted file that stores certificates, is generated in the directory <*UMP_installation*>/Nimsoft/probes/service/wasp/conf
- A 1024-bit self-signed certificate is automatically generated in the wasp.keystore

At this point, if you only require a 1024-bit self-signed certificate, there are no further steps you must complete. If you require any other certificate, you must obtain that certificate, and then replace the automatically generated 1024-bit self-signed certificate. The instructions in each scenario provide the specific steps to do so.

Important! If you require a certificate other than the automatically generated 1024-bit self-signed certificate described above, ensure that you read and understand the section The ssl_reinitialize_keystore Callback (see page 9).

The ssl_reinitialize_keystore Callback

As described in the previous section, a 1024-bit self-signed certificate is automatically generated in the wasp.keystore when you enable HTTPS in the wasp.cfg file. *To use any other certificate, you will be required to enter a valid password for the wasp.keystore*; however, the wasp.keystore has a *hard-coded, unknown* password. Therefore, the first time you configure the wasp for SSL, it is recommended you execute the *ssl_reinitialize_keystore* callback and set a new password.

The ssl_reinitialize_keystore callback re-creates the wasp.keystore and its password hash. When you run this callback, enter a new password as an argument, and then *securely store the new password for future use*. If you lose or forget this password, the only way to reset it is to reinitialize the wasp.keystore again.

Important! Use caution with the ssl_reinitialize_keystore callback. This callback changes the encryption hash of the wasp.keystore, and will *invalidate any certificates you are currently using*. For this reason, it is strongly recommended that you back up individual key and certificate files, so that if you have to reinitialize the keystore, you can reload the keys and certificates into the new keystore.

In addition, do not use the keytool utility to change the password of the wasp.keystore, as the wasp will not recognize the new password. Currently, the only way to change the password of the wasp.keystore is to use the ssl_reinitialize_keystore callback.

Use Cases for SSL with UMP

Nimsoft recommends that you consult your network security engineers and compliance specialists regarding your specific security requirements before using this guide.

In general, industry-standard security requirements mandate the use of SSL encryption for client-server communications via an untrusted network. This includes the following situations:

- If users access UMP via a public network, such as the Internet
- If sessions traverse an unsecured part of your network, such as wireless networks in meeting rooms or in public-access areas
- If sessions traverse mobile networks

Note: For high-security environments, it is recommended that you use at least 2048-bit encryption. However, bear in mind that longer RSA key-lengths significantly affect the speed of encryption, and of decryption in particular.

Prerequisites

Before using the scenarios in this document, ensure that you meet the following prerequisites:

- Your environment is configured to run keytool commands if you plan to use a certificate other than a 1024-bit self-signed certificate. This means that the $PATH system variable includes a path to java.exe and keytool. See the section keytool Command Not Found (see page 33) for additional information.
- You are familiar with public key infrastructure (PKI) and system administration.

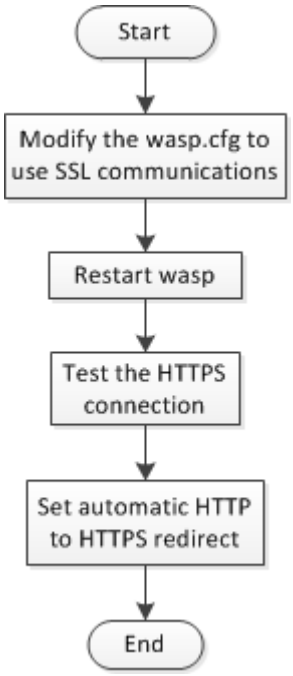
Additional SSL Resources

In addition to the appendix Troubleshooting SSL Certificates (see page 31), the following websites provide helpful tools and resources for setting up and managing SSL certificates:

- http://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/keytool.html
- www.sslshopper.com

Chapter 2: Implement a 1024-bit Self-Signed SSL Certificate

This chapter provides instructions for configuring UMP to use a 1024-bit self-signed SSL certificate. The high-level steps for implementing a 1024-bit self-signed SSL certificate are shown in the following flow drawing.



The steps in the above drawing correspond to the steps in the following section, 1024-bit Self-Signed Certificate Procedure (see page 11).

1024-bit Self-Signed Certificate Procedure

This section describes how to modify the wasp.cfg file in order to use SSL communications with UMP. When these changes take effect, a *1024-bit self-signed certificate is automatically generated and stored in the wasp.keystore*.

Note: The automatically generated 1024-bit SSL certificate has a validity period of one year.

1. Modify the wasp.cfg to use SSL communications:
   a. Open Infrastructure Manager.
   b. Navigate to the server running UMP, and locate the wasp probe.
   c. Press the <Ctrl> key as you right-click the wasp probe, and then select **Raw Configure**.
   d. With the **setup** section highlighted, locate the **https_port** key, and click **Edit Key** to specify a port. If necessary, click **New Key** and enter **https_port**.

      Note: The maximum port value you can set is 65535.
   e. Edit the **https_max_threads** key to configure the number of concurrent https requests.

      The default value is 500.

      After you click **OK**, the wasp is configured to use SSL. The first time the wasp starts with SSL enabled, a new keystore, *wasp.keystore*, is generated and stored in <*UMP_installation*>/probes/service/wasp/conf. In addition, a 1024-bit self-signed certificate is generated and stored in the wasp.keystore file.

      Note: If you are using Mozilla Firefox as your browser with a self-signed certificate, you must set the wasp configuration variable webapps/sdp/use_html_upload=1 to upload images and dashboards. This is due to limitations of the Adobe FlashPlayer plug-in.
2. Restart wasp.

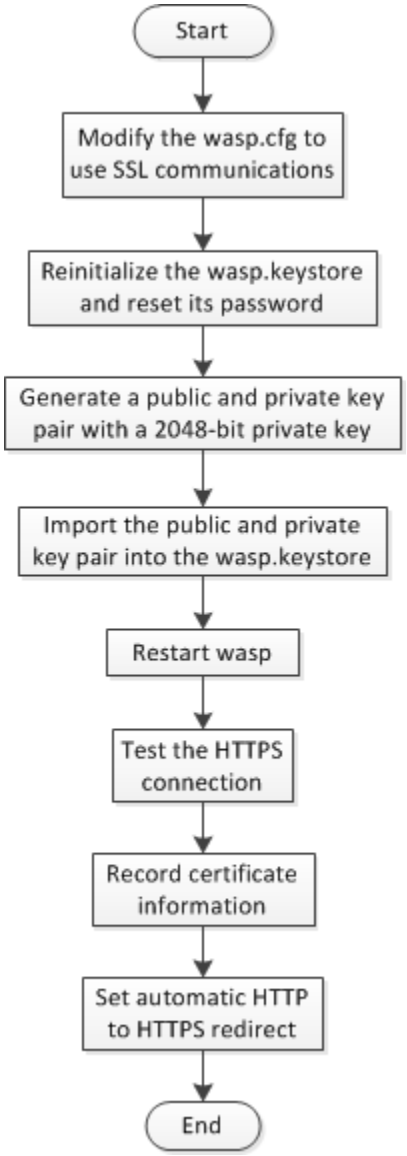
   The wasp is now configured to use SSL with a 1024-bit self-signed certificate.
3. Test the HTTPS connection:
   a. Verify that you can now access UMP using HTTPS.
   b. Click the lock icon to the left of the URL in the browser address window to view information about the connection.
4. Set automatic HTTP to HTTPS redirect:
   a. Locate the following directory:

      <*Nimsoft_installation*>/Nimsoft/probes/service/wasp/webapps/ROOT/WEB-INF/classes.
   b. Open the file portal-ext.properties in a text editor.
   c. At the bottom of the portal-ext.properties file, add the line web.server.protocol=https.
   d. Save the portal-ext.properties file and restart the wasp probe.

      UMP is now configured to redirect an HTTP login attempt to HTTPS.

Chapter 3: Implement a 2048-bit Self-Signed SSL Certificate

This chapter provides instructions for configuring UMP to use a 2048-bit self-signed SSL certificate. The high-level steps for implementing a 2048-bit self-signed SSL certificate are shown in the following flow drawing.



The steps in the above drawing correspond to the steps in the following section, 2048-bit Self-Signed Certificate Procedure (see page 14).

2048-bit Self-Signed Certificate Procedure

Follow these steps:

1. Modify the wasp.cfg to use SSL communications:
   a. Open Infrastructure Manager.
   b. Navigate to the server running UMP, and locate the wasp probe.
   c. Press the <Ctrl> key as you right-click the wasp probe, and then select **Raw Configure**.
   d. With the **setup** section highlighted, locate the **https_port** key, and click **Edit Key** to specify a port. If necessary, click **New Key** and enter **https_port**.

      Note: The maximum port value you can set is 65535.
   e. Edit the **https_max_threads** key to configure the number of concurrent https requests.

      The default value is 500.

      After you click **OK**, the wasp is configured to use SSL. The first time the wasp starts with SSL enabled, a new keystore, *wasp.keystore*, is generated and stored in <*UMP_installation*>/probes/service/wasp/conf. In addition, a 1024-bit self-signed certificate is generated and stored in the wasp.keystore file.

      Note: If you are using Mozilla Firefox as your browser with a self-signed certificate, you must set the wasp configuration variable webapps/sdp/use_html_upload=1 to upload images and dashboards. This is due to limitations of the Adobe FlashPlayer plug-in.

2. Reinitialize the wasp.keystore and reset its password.

   Important! Perform the following sub-steps *only if at least one of the following statements is true*:

   - You do not know the password for the wasp.keystore.
   - This is the *first time* you are configuring UMP to use SSL.

   If neither of the above statements is true, read and understand the section The ssl_reinitialize_keystore Callback (see page 9) before using the following sub-steps.

   a. Open Infrastructure Manager.

   b. Navigate to the server running UMP.

   c. Click on the wasp probe to highlight it.

   d. Press <Ctrl>+<P> to open the probe utility.

   e. In the drop-down menu under **Probe commandset**, select **ssl_reinitialize_keystore**.

   f. Enter a new password as an argument.

      Note: Use a password that is at least six characters long. The wasp probe utility will not prevent you from using a shorter password, but you will be unable to make changes to the wasp.keystore as described later in these steps otherwise.

   g. Click the green play button to run the callback.

      The **Command** status bar displays the text **OK**.

3. Generate a public and private key pair with a 2048-bit private key:

   a. Open an administrator command prompt on the server running UMP.

      Note: Run the following keytool commands in the same directory as the wasp.keystore file, typically <*UMP_installation*>/probes/service/wasp/conf.

      The keytool utility is located in the directory where the JRE resides, typically <*UMP_installation*>/jre/<*jre_version*>/bin/keytool.

   b. Verify that you have a valid password for the wasp.keystore:
      ```
      <UMP_installation>/jre/<jre_version>/bin/keytool -list
      -keystore wasp.keystore
      ```

   c. Delete the automatically generated private key:
      ```
      <UMP_installation>/jre/<jre_version>/bin/keytool -delete
      -alias wasp -keystore wasp.keystore
      ```

   d. Verify the key was deleted:
      ```
      <UMP_installation>/jre/<jre_version>/bin/keytool -list
      -keystore wasp.keystore
      ```

e. Generate the public and private key pair with a 2048-bit private key:
`<UMP_installation>/jre/<jre_version>/bin/keytool -genkeypair -alias wasp -keyalg RSA -keysize 2048 -keystore wasp.keystore –validity <days_cert_is_valid>`

f. When prompted for your first and last name, enter the FQDN.

g. When prompted, provide entries for the following:

- Organizational unit
- Organization
- City or Locality
- State or Province
- Two-letter country code

You are prompted to confirm that the information you entered is correct.

4. Import the public and private key pair into the wasp.keystore:
`<UMP_installation>/jre/<jre_version>/bin/keytool –import –trustcacerts –alias wasp –file <my_domain>.crt –keystore wasp.keystore`

5. Restart wasp.

The wasp is now configured to use SSL with a 2048-bit self-signed certificate.

6. Test the HTTPS connection:

a. Verify that you can now access UMP using HTTPS.

b. Click the lock icon to the left of the URL in the browser address window to view information about the connection.

7. Record certificate information:

a. If you used the ssl_reinitialize_keystore callback, securely record the new password you set for the wasp.keystore.

b. Ensure that you record the validity period you set for the certificate.

c. Back up the certificate files to a secure location.

8. Set automatic HTTP to HTTPS redirect:
    a. Locate the following directory:

        <*Nimsoft_installation*>/Nimsoft/probes/service/wasp/webapps/ROOT/WEB-INF/classes.

    b. Open the file portal-ext.properties in a text editor.

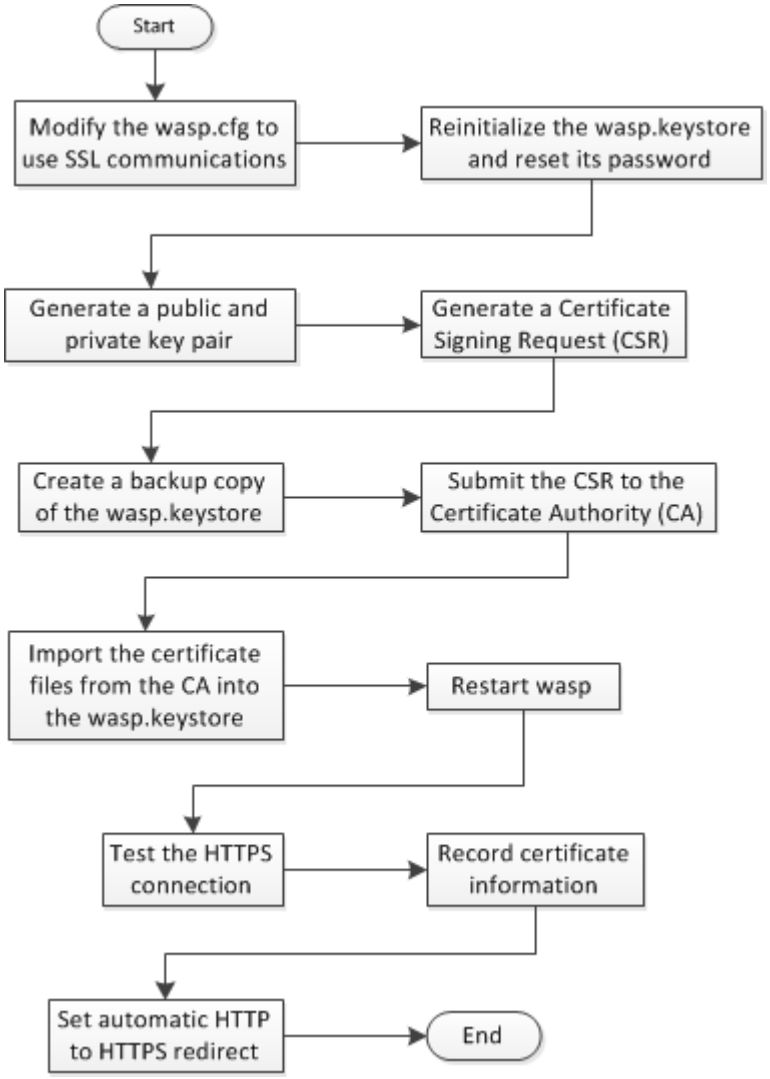
    c. At the bottom of the portal-ext.properties file, add the line web.server.protocol=https.

    d. Save the portal-ext.properties file and restart the wasp probe.

        UMP is now configured to redirect an HTTP login attempt to HTTPS.

Chapter 4: Implement a CA-Signed SSL Certificate

This chapter provides instructions for configuring UMP to use an SSL certificate signed by a certificate authority (CA). The high-level steps for implementing a CA-signed certificate are shown in the following flow drawing.



The steps in the above drawing correspond to the steps in the section CA-Signed Certificate Procedure (see page 21).

Entity, Intermediate, and Root Certificates

A number of CAs issue intermediate, or *chained* certificates. If your CA issues chained certificates, you will typically receive the following certificate files:

- An *entity* certificate
- One or more *intermediate* certificates
- A root certificate may be included

You must upload the entity certificate and any intermediate certificates your CA provides. You may not need to upload a root certificate. This is because the NMS installation automatically installs a Java Runtime Environment (JRE) that includes the root certificates of many CAs. However, your CA may provide a new root certificate and advise that you upload it.

View Root Certificates

You can view the root certificates installed automatically with the JRE during the NMS installation.

Follow these steps:

1. Open an administrator command prompt on the server running UMP.
2. Change directories as follows:
   `cd <UMP_installation>/jre/<jre_version>/lib/security`
3. Issue the following command:
   `<UMP_installation>/jre/<jre_version>/bin/keytool keytool -list -keystore cacerts`

   The system prompts you to enter the keystore password. After you enter a valid password, the system displays the default root certificates in the cacerts file.

CA-Signed SSL Certificate Procedure

Follow these steps:

1. Modify the wasp.cfg to use SSL communications:
   a. Open Infrastructure Manager.
   b. Navigate to the server running UMP, and locate the wasp probe.
   c. Press the <Ctrl> key as you right-click the wasp probe, and then select **Raw Configure**.
   d. With the **setup** section highlighted, locate the **https_port** key, and click **Edit Key** to specify a port. If necessary, click **New Key** and enter **https_port**.

      Note: The maximum port value you can set is 65535.
   e. Edit the **https_max_threads** key to configure the number of concurrent https requests.

      The default value is 500.

      After you click **OK**, the wasp is configured to use SSL. The first time the wasp starts with SSL enabled, a new keystore, *wasp.keystore*, is generated and stored in <*UMP_installation*>/probes/service/wasp/conf. In addition, a 1024-bit self-signed certificate is generated and stored in the wasp.keystore file.

2. Reinitialize the wasp.keystore and reset its password.

   Important! Perform the following sub-steps *only if at least one of the following statements is true*:

   - You do not know the password for the wasp.keystore.
   - This is the *first time* you are configuring UMP to use SSL.

   If neither of the above statements is true, read and understand the section The ssl_reinitialize_keystore Callback (see page 9) before using the following sub-steps.

   a. Open Infrastructure Manager.

   b. Navigate to the server running UMP.

   c. Click on the wasp probe to highlight it.

   d. Press <Ctrl>+<P> to open the probe utility.

   e. In the drop-down menu under **Probe commandset**, select **ssl_reinitialize_keystore**.

   f. Enter a new password as an argument.

      Note: Use a password that is at least six characters long. The wasp probe utility will not prevent you from using a shorter password, but you will be unable to make changes to the wasp.keystore as described later in these steps otherwise.

   g. Click the green play button to run the callback.

      The **Command** status bar displays the text **OK**.

   h. Securely record the password you set for future use.

3. Generate a public and private key pair:

   a. Open an administrator command prompt on the server running UMP.

      Note: Run the following keytool commands in the same directory as the wasp.keystore file, typically <*UMP_installation*>/probes/service/wasp/conf.

      The keytool utility is located in the directory where the JRE resides, typically <*UMP_installation*>/jre/<*jre_version*>/bin/keytool.

   b. Verify that you have a valid password for the wasp.keystore:
      ```
      <UMP_installation>/jre/<jre_version>/bin/keytool -list
      -keystore wasp.keystore
      ```

   c. Delete the automatically generated private key:
      ```
      <UMP_installation>/jre/<jre_version>/bin/keytool -delete
      -alias wasp -keystore wasp.keystore
      ```

   d. Verify the key was deleted:
      ```
      <UMP_installation>/jre/<jre_version>/bin/keytool -list
      -keystore wasp.keystore
      ```

e. Generate the public and private key pair with the key size you require:
`<UMP_installation>/jre/<jre_version>/bin/keytool -genkeypair -alias wasp -keyalg RSA -keysize <key_size> -keystore wasp.keystore –validity <days_cert_is_valid>`

f. When prompted for your first and last name, enter the FQDN.

g. When prompted, provide entries for the following:

- Organizational unit
- Organization
- City or Locality
- State or Province
- Two-letter country code

You are prompted to confirm that the information you entered is correct.

4. Generate a Certificate Signing Request (CSR):
`<UMP_installation>/jre/<jre_version>/bin/keytool -certreq -alias wasp -validity <days_cert_is_valid> -keystore wasp.keystore -file <your_domain>.csr`

5. Create a backup copy of the wasp.keystore.

Note: This is not a required step, but it is strongly recommended. In the event you encounter a problem later in this procedure, a backup copy of the wasp.keystore will save you from having to repeat previous steps.

6. Submit the CSR to the Certificate Authority (CA):

a. Paste the CSR into the web form of the CA.

b. Remove any characters before **----BEGIN CERTIFICATE REQUEST** and after **END CERTIFICATE REQUEST----.**

7. Import the certificate files from the CA into the wasp.keystore:

   Note: All keystore entries must use a unique alias. You must use the alias *wasp* for the signed, or entity certificate. If your CA provides multiple intermediate certificates, each intermediate certificate must also use a unique alias.

   a. If your CA provided a root certificate, import the root certificate:
   `<UMP_installation>/jre/<jre_version>/bin/keytool -import -trustcacerts -alias <root_certificate> -file <root_certificate>.cer -keystore wasp.keystore`

   b. Import the first intermediate certificate:
   `<UMP_installation>/jre/<jre_version>/bin/keytool -import -trustcacerts -alias <first_intermediate_certificate> -file <first_intermediate_certificate>.cer -keystore wasp.keystore`

   c. Repeat the previous step for each additional intermediate certificate.

   d. Import the signed certificate. This is the entity certificate if you received a chained certificate:
   `<UMP_installation>/jre/<jre_version>/bin/keytool –import –trustcacerts –alias wasp –file <your_domain>.crt –keystore wasp.keystore`

8. Restart wasp.

   The wasp is now configured to use an SSL connection with a CA-signed certificate.

9. Test the HTTPS connection:

   a. Verify that you can now access UMP using HTTPS.

   b. Click the lock icon to the left of the URL in the browser address window to view information about the connection.

10. Record certificate information:

    a. If you used the ssl_reinitialize_keystore callback, securely record the new password you set for the wasp.keystore.

    b. Ensure that you record the validity period you set for the certificate.

    c. Back up the certificate files to a secure location.

11. Set automatic HTTP to HTTPS redirect:

    a. Locate the following directory:

    <*Nimsoft_installation*>/Nimsoft/probes/service/wasp/webapps/ROOT/WEB-INF/classes.

    b. Open the file portal-ext.properties in a text editor.

    c. At the bottom of the portal-ext.properties file, add the line web.server.protocol=https.

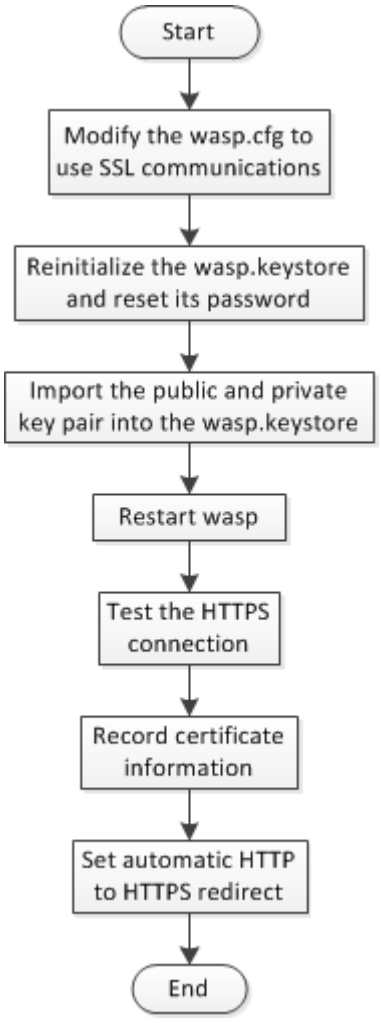
    d. Save the portal-ext.properties file and restart the wasp probe.

    UMP is now configured to redirect an HTTP login attempt to HTTPS.

Chapter 5: Implement a Wildcard SSL Certificate

This chapter describes how to configure UMP to use a wildcard SSL certificate. Wildcard SSL certificates allow you to secure your domain and an unlimited number of sub-domains, using a single SSL certificate, *.<*your_domain*>.com.

The high-level steps for implementing a wildcard SSL certificate are shown in the following flow drawing.



The steps in the above drawing correspond to the steps in the following section, Wildcard Certificate Procedure (see page 28).

Wildcard Certificate Procedure

After obtaining a wildcard certificate, use the steps in this section to configure UMP.

Follow these steps:

1. Modify the wasp.cfg to use SSL communications:
   a. Open Infrastructure Manager.
   b. Navigate to the server running UMP, and locate the wasp probe.
   c. Press the <Ctrl> key as you right-click the wasp probe, and then select **Raw Configure**.
   d. With the **setup** section highlighted, locate the **https_port** key, and click **Edit Key** to specify a port. If necessary, click **New Key** and enter **https_port**.

      Note: The maximum port value you can set is 65535.
   e. Edit the **https_max_threads** key to configure the number of concurrent https requests.

      The default value is 500.

      After you click **OK**, the wasp is configured to use SSL. The first time the wasp starts with SSL enabled, a new keystore, *wasp.keystore*, is generated and stored in <*UMP_installation*>/probes/service/wasp/conf. In addition, a 1024-bit self-signed certificate is generated and stored in the wasp.keystore file.

2. Reinitialize the wasp.keystore and reset its password.

   Important! Perform the following sub-steps *only if at least one of the following statements is true*:

   - You do not know the password for the wasp.keystore.
   - This is the *first time* you are configuring UMP to use SSL.

   If neither of the above statements is true, read and understand the section The ssl_reinitialize_keystore Callback (see page 9) before using the following sub-steps.

   a. Open Infrastructure Manager.

   b. Navigate to the server running UMP.

   c. Click on the wasp probe to highlight it.

   d. Press <Ctrl>+<P> to open the probe utility.

   e. In the drop-down menu under **Probe commandset**, select **ssl_reinitialize_keystore**.

   f. Enter a new password as an argument.

      Note: Use a password that is at least six characters long. The wasp probe utility will not prevent you from using a shorter password, but you will be unable to make changes to the wasp.keystore as described later in these steps otherwise.

   g. Click the green play button to run the callback.

      The **Command** status bar displays the text **OK**.

   h. Securely record the password you set for future use.

3. Import the public and private key pair into the wasp.keystore:

   a. Open an administrator command prompt on the server running UMP.

      Note: Run the following keytool commands in the same directory as the wasp.keystore file, typically <*UMP_installation*>/probes/service/wasp/conf.

      The keytool utility is located in the directory where the JRE resides, typically <*UMP_installation*>/jre/<*jre_version*>/bin/keytool.

   b. Issue the following keytool command to verify the wasp alias:
      `<Nimsoft_installation>/jre/<jre_version>/bin/keytool -list -alias wasp -keystore wasp.keystore`

   c. Issue the following command to import the certificate and keys:
      `<Nimsoft_installation>/jre/<jre_version>/bin/keytool -import -keystore wasp.keystore -srckeystore <my_keystore>.p12 -srcstoretype PKCS12 -alias wasp`

   d. Choose **yes** at the prompt **Existing entry alias wasp exists, overwrite?**

   e. Issue the following command to verify that the wasp.keystore was updated:
      `<Nimsoft_installation>/jre/<jre_version>/bin/keytool -list -keystore wasp.keystore`

4. Restart the wasp.

   The wasp is now configured to use a wildcard certificate.

5. Test the HTTPS connection:

   a. Verify that you can now access UMP using HTTPS.

   b. Click the lock icon to the left of the URL in the browser address window to view information about the connection.

6. Record certificate information:

   a. If you used the ssl_reinitialize_keystore callback, securely record the new password you set for the wasp.keystore.

   b. Ensure that you record the validity period you set for the certificate.

   c. Back up the certificate files to a secure location.

7. Set automatic HTTP to HTTPS redirect:

   a. Locate the following directory:

      <*Nimsoft_installation*>/Nimsoft/probes/service/wasp/webapps/ROOT/WEB-INF/classes.

   b. Open the file portal-ext.properties in a text editor.

   c. At the bottom of the portal-ext.properties file, add the line web.server.protocol=https.

   d. Save the portal-ext.properties file and restart the wasp probe.

      UMP is now configured to redirect an HTTP login attempt to HTTPS.

Appendix A: Troubleshooting SSL Certificates

This appendix provides information to help you troubleshoot issues implementing SSL with UMP.

This section contains the following topics:

Alias <wasp> Already Exists

Symptom:

I see the exception:

```
java.lang.Exception: Key pair not generated, alias <wasp> already exists
```

Solution:

All keystore entries must have a unique alias. When you configure the wasp.cfg to enable SSL, 1024-bit self-signed certificate using the alias wasp is automatically generated in the wasp.keystore. To use a different certificate, you must delete this keystore entry first.

Issue the following keytool command in the same directory as the wasp.keystore:
<UMP_installation>`/jre/`*<jre_version>*`/bin/keytool -delete -alias wasp -keystore wasp.keystore`

Alias Name wasp Does Not Identify a Key Entry

Symptom:

I see the exception:

```
java.io.IOException: Alias name wasp does not identify a key entry
```

Solution:

This exception may occur if you generated a CSR using Microsoft Internet Information Services (IIS). If you use IIS, the certificate and keys you obtain from the CA may not be in a format that the wasp.keystore can import. In this case, you must convert the certificate files to the PKCS#12, or PFX format before importing them.

Note: The following requires O*penSSL,* a library that provides cryptographic functionality. You can obtain binary distributions at http://www.openssl.org/related/binaries.html.

Issue the following openssl command to convert the certificate to the PFX format:
`openssl pkcs12 -export -out <pfx_file>.pfx -inkey <private_key>.key -in <cert_file>.crt -certfile CACert.crt`

See the website https://www.sslshopper.com for additional help with converting certificate files.

Given Final Block Exception

Symptom:

I see the exception:

```
javax.crypto.BadPaddingException: Given final block not properly padded
```

Solution:

Note: The following requires O*penSSL,* a library that provides cryptographic functionality. You can obtain binary distributions at http://www.openssl.org/related/binaries.html.

Issue the following OpenSSL commands to overwrite the existing wasp alias in the keystore:

```
openssl pkcs12 -in <my_pfx_file>.pfx -out <my_pem_file>.pem
openssl pkcs12 -export -in <my_pem_file> -out <my_keystore>.p12 -name wasp
```

keytool Command Not Found

Symptom:

When I issue a keytool command, a message tells me the command was not found.

Solution:

Verify that paths are set for java.exe and keytool in the $PATH system variable:

1. Open an administrator command prompt on the server running UMP.
2. Issue the following command in the same directory as the wasp.keystore, typically <*UMP_installation*>/probes/service/wasp/conf:
   `java -version`
3. If the system returns errors instead of java version information, add paths for java.exe and keytool to the $PATH system variable.

Signer Cert Does Not Match Issuer Name

Symptom:

I see the exception:

```
java.security.cert.CertificateException: Subject name of signer cert
does not match issuer name of supplied cert chain
```

Solution:

This or a similar exception may occur if your CA issued a *chained* certificate, but the intermediate certificate(s) was not uploaded. You must upload the entity certificate *and* any intermediate certificates your CA provides.

Issue the following keytool command in the same directory as the wasp.keystore, typically <*UMP_installation*>/probes/service/wasp/conf:

```
<UMP_installation>/jre/<jre_version>/bin/keytool -import -keystore
wasp.keystore -trustcacerts -file <intermediate_cert>.CER
```

Note: All keystore entries must use a unique alias. You must use the alias *wasp* for the signed, or entity certificate. If your CA provides multiple intermediate certificates, each intermediate certificate must also use a unique alias.

See the section Entity, Intermediate, and Root Certificates (see page 20) for additional information.