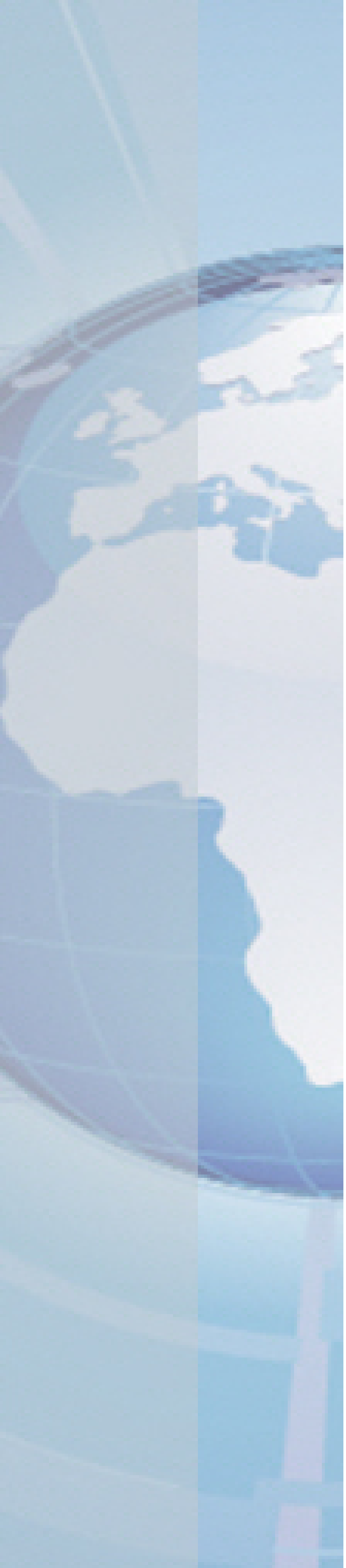

Whitepaper

Authenticating and Encrypting Communication between LISTSERV® Maestro Components

August 25, 2010

Information in this document is subject to change without notice. Companies, names, and data used for example herein are fictitious unless otherwise noted. Some screen captures have been cropped and/or edited for emphasis or descriptive purposes.

L-Soft invites comments on its documentation. Please feel free to send your comments by email to: manuals@lsoft.com

Introduction

The purpose of this white paper is to describe how to use Secure Sockets Layer (SSL) to authenticate and encrypt communication between components of a LISTSERV Maestro installation. In order to keep this paper concise and not repeat passages of existing documentation, there are many references to various sections of the *LISTSERV Maestro Administrator's Manual*. These sections provide necessary background information and instructions.

Using SSL to authenticate and encrypt the communication between components is different than using SSL to secure communication between the user and the Maestro User Interface or Administration Hub (HTTPS). However, the background information, and many of the steps to use SSL for both these applications, is very similar. Section 18, Securing Access with SSL, of the *LISTSERV Maestro Administrator's Manual* describes how to use SSL to secure communication between the user and LISTSERV Maestro components. Understanding this information is key to implementing SSL for inter-component communication. With this knowledge, it is also possible to combine both applications of SSL within a LISTSERV Maestro installation.

LISTSERV Maestro's Internal Communication

A LISTSERV Maestro installation consists of several components: the Maestro User Interface (LUI), the Administration Hub (HUB), and Maestro Tracker (TRK). These components communicate with each other over network sockets, which are connected to ports. This happens using the “internal communications port” (default: port 1099) and the “tracker communications port” (default: port 7000).

These ports are opened for incoming connections on the servers where the components are installed. This is described in Section 14, Using Non-Standard Ports, of the *LISTSERV Maestro Administrator's Manual*. The ports can be protected against unauthorized access by configuring the firewall that protects the server(s) as described in Section 16, Installing behind a Firewall, of the *LISTSERV Maestro Administrator's Manual*.

For stricter security requirements that ensure no unauthorized access to these ports from anyone, either inside or outside the firewall, Secure Sockets Layer can be used on these ports. SSL acts to authenticate and encrypt communication on ports opening to receive data from the outside. Even if all the components of an installation are on the same server, communication between ports can be authenticated using SSL. The encryption feature of SSL gives your installation additional security by encrypting communication between ports. This is valuable when components are located on different machines and communication has to travel across the network.

General Considerations

Before you continue, please read Section 18, Securing Access with SSL, of the *LISTSERV Maestro Administrator's Manual*. Its general discussion of authentication, encryption, and certificates applies in the same manner to using SSL for the inter-component communication. The remaining sections of this white paper assume that you understand the role of server certificates and trusted root certificates.

Securing the inter-component communication is an “all-or-nothing” undertaking. You can either secure all communication on the “internal communication port” and the “tracker communication port” between all three components (LUI, HUB, and TRK) or none of it. You cannot secure the communication between some components or on some ports and not between all the others.

Important: Because securing inter-component communication is independent of securing user access by HTTPS, you can use SSL to secure inter-component communication, user access, both of them, or none of them, depending on your security needs.

For any given LISTSERV Maestro installation, there may be one, two, or three servers, depending on how the components are distributed. Because secure inter-component communication is “all-or-nothing," all of those servers need to be secured (or all of them remain unsecured). Securing inter-component communication happens on a per-server basis. It does not matter how many of the LISTSERV Maestro components are installed on one server. For the purpose of securing the communication, they are considered as a single server entity, and when secured, all the communication of all components installed there will be secured. For this reason, throughout the rest of this document, individual LISTSERV Maestro components will not be named; only the servers they reside on will be referenced.

Important: Securing the inter-component communication deals only with the communication on the described ports between the three main components LUI, HUB and TRK. Communication to the system database or any user database and communication to LISTSERV and LSMTP is not part of this discussion.

Obtaining and Installing Server Certificates

A fully authenticated secure communication using SSL is divided into two parts: first, the authentication that the two communication partners really are the entities they claim they are, and second, the encryption of all data passed between the two.

In the SSL protocol, the two partners first authenticate each other by presenting their signed server certificates to each other. Each partner then checks the signature of the other’s certificate to see if it has been signed by a trusted certificate authority (CA). Once the two partners have authenticated each other, they then agree on an encryption that they then employ for all subsequent communication. As a consequence, to have authenticated SSL communication between two servers, both servers must possess a server certificate that has been signed by a certificate authority (CA) and that the other server accepts as a trusted authority.

The first step in securing the inter-component communication is to obtain signed server certificates for all LISTSERV Maestro servers. Obtaining and installing these certificates is very similar to the steps described for the certificate that is required to secure user-access with HTTPS in Section 18, Securing Access with SSL of the *LISTSERV Maestro Administrator's Manual*. To avoid repeating the description of the required steps here, references to the appropriate sub-sections of manual are included.

Before we launch into the detailed steps of obtaining and installing a certificate, some additional issues need to be considered:

- You need to obtain a signed server certificate for all LISTSERV Maestro servers, one for each server.
    - The certificate is always bound to the explicit server name that you chose when you created the certificate.
    - You cannot simply obtain any server certificate and use it on a server of your choice.
    - The same certificate cannot be used on several servers.
    - If your server were renamed, you would have to obtain a new certificate for the new name.

- The certificate must match the name that the server uses to identify itself during network communication.
  - It is possible that a server has more than one name assigned to it. The name that a server uses to identify itself during network communication may not be the one you expect.
  - If the server certificate was created with one name but the server identifies itself with another name, the SSL communication will fail because the communication partner cannot match the certificate to the server.
  - For example, a server may have several names assigned to it for communication with different applications and servers. The server might use one name to identify itself to communication partners that reside on the same server, a second name for partners on different servers but in the same network zone, and a third name for servers in a different zone. If this is the case, you need to consolidate your server so that it uses the same name for all these different types of connections, or you must obtain and install one certificate for each of these names. Using several certificates for the same server with different server-names is an untested feature – test and employ at your own risk.
- We recommend that you obtain all server certificates from the same CA so that you will have to deal with only one trusted root authority certificate.
- If you already have obtained and installed (in a keystore file of your choosing) a signed certificate for the purpose of securing user access to a server with HTTPS (Section 18, Securing Access with SSL) then you do not need to obtain another certificate for this server. The same certificate can also be used for securing the inter-component communication on that server. Simply supply the path and password of your existing keystore file when performing the step described in the *Enabling Secure Inter-Component Communication* section.

Obtaining a server certificate involves three basic steps. The steps are explained in more detail in the remaining sections of this document. The steps are:

1. Creating an unsigned certificate with the name of the server you want to secure.
2. Creating a certificate signing request (CSR) from that certificate and sending it to a certification authority (CA). The CA first verifies that you really are who you claim to be and then returns a signed version of your certificate to you.
3. Replacing the unsigned certificate with the signed certificate you received back from the CA.

The administration of the certificates happens with a command line tool called "`keytool`" that is installed together with Java. For more information about this tool, and further discussion about certificates and secure communication, see the documentation available from the Sun Microsystems Web site:

http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html

and

http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html

Important: The steps described in the following sections for obtaining and installing a server certificate must be executed for each server that is to be secured (for each server where any of the three LISTSERV Maestro main components LUI, HUB, or TRK is installed).

Securing the Trusted Root Certificate Keystore

As a first step on each server, secure the default keystore for trusted root certificates that is shipped with Java. See Section 18.3.1, Securing the Trusted Root Certificate Keystore, of the *LISTSERV Maestro Administrator's Manual* for details.

Creating an Unsigned Server Certificate

On each server, create an unsigned certificate with the name of this server. See Section 18.3.2, Creating an Unsigned Server Certificate, of the *LISTSERV Maestro Administrator's Manual* for details.

Performing a Certificate Signing Request (CSR)

On each server, generate a certificate signing request (CSR) from the unsigned server certificate and submit it to a CA of your choice, for example VeriSign. See Section 18.3.3, Performing a Certificate Signing Request (CSR), of the *LISTSERV Maestro Administrator's Manual* for details.

Installing the Signed Server Certificate

On each server, replace the unsigned version of the server certificate that you have in your keystore file with the signed version that you received back from your CA. See Section 18.3.4, Installing the Signed Server Certificate, of the *LISTSERV Maestro Administrator's Manual* for details.

If during the import of the signed certificate you get an error message that no trusted root certificate can be found, you first need to import the root certificate for your CA (see the next section) and then repeat the step of importing your signed certificate as described above.

Installing a Trusted Root Certificate

If you did not get an error message in the previous step saying that no trusted root certificate can be found, skip to the *Enabling Secure Inter-Component Communication* section.

Trusted root certificates are required for two purposes:

- **To import a signed server certificate:**

  When you receive your signed server certificate back from your CA and want to import it into the keystore of your server, you can only do so if the Java that comes with LISTSERV Maestro already recognizes this CA as a trusted CA. In order for the system to trust the signed certificate, it first must trust the signer.

  The Java installed with LISTSERV Maestro comes with certificates of a few trusted CAs (such as VeriSign and Thawte) pre-installed.

  If your CA is not listed as a trusted CA, you must first import your CA's root certificate into LISTSERV Maestro's default keystore of trusted certificates on the server where you want to import the signed server certificate. Most likely, this will also have to be done on all other LISTSERV Maestro servers involved.

- **To authenticate a communication partner:**

  During SSL communication, certificates are used to authenticate the communication partners. This happens in a way that is most easily explained with an example:

  Assume that we have two servers communicating with each other, one called "Alice" and the other called "Bob". For simplicity's sake, we are only looking at the authentication process in one direction - how Bob authenticates the identity of Alice. In reality, this happens in just the same fashion in the other direction - how Alice authenticates Bob's identity.

Alice has previously obtained a server certificate from a CA called "TrustCorp". This "Alice-certificate" has been signed with the CA's own certificate, the "TrustCorp-certificate".

At the beginning of the communication, Alice sends her own Alice-certificate (which is signed with the TrustCorp-certificate) to Bob.

Bob verifies the Alice-certificate by checking the signature that has been made with the TrustCorp-certificate. To do this, Bob must be able to match the TrustCorp-certificate with a trusted certificate in his own trusted root certificate keystore. Bob must already be aware of Alice's CA's root-certificate, the TrustCorp-certificate, as a trusted certificate to actually be able to verify Alice's certificate.

For LISTSERV Maestro, this means that if a certain CA has been used to sign the certificate of a given LISTSERV Maestro server, then all other LISTSERV Maestro servers must be aware that this CA is a trusted CA (or better that the certificate of this CA is a trusted root certificate).

If the CA's certificate is one of the pre-defined trusted root certificates of the Java that is installed with LISTSERV Maestro, you need to import it as a trusted certificate on all servers, not only on the server with the certificate that was signed by this CA.

This is also the reason why it is a good idea to use the same CA to sign all server certificates. That way you only need to import one and the same root certificate on all servers, a maximum of three imports. If instead you have a different CA for each server, then you may need to import the root certificate of each CA on each server, a maximum of nine imports.

See Section 18.3, Obtaining and Installing a Server Certificate, of the *LISTSERV Maestro Administrator's Manual* for details about how to obtain/install the trusted root certificate of a CA.

Again, this step is only required if the root certificate of the CA of your choice is not among those that are pre-installed in the Java that comes with LISTSERV Maestro. This becomes apparent when you try to import your signed certificate. If the import is successful, your CA is among the trusted ones. If your import fails, you will have to import the CA's root certificate. In the case of a failed import, this step needs to be executed for each "unknown" CA, and on all LISTSERV Maestro servers, not only on the server with the certificate signed by this CA.

Once the root certificate has been installed on all the servers, return to the *Installing the Signed Server Certificate* section.

Enabling Secure Inter-Component Communication

After you have successfully obtained and installed server certificates for all LISTSERV Maestro servers, you need to perform one final administration step to actually enable secure inter-component communication:

On each LISTSERV Maestro server, create an INI-file with the following path and name:

```
\Program Files\L-Soft\Application Server\commands\ssl.ini
```

The filename is in lowercase: "`ssl.ini`"

This file must be a text file created according to the LISTSERV Maestro INI-file rules in Section 20, Editing LISTSERV Maestro INI Files, of the *LISTSERV Maestro Administrator's Manual*. There are exactly the following three entries for this INI file:

```
SSLAllowedClients=CLIENT_LIST
SSLKeystorePath=KEYSTORE_FILE
SSLKeystorePassword=PASSWORD
```

Use the following replacements:

CLIENT_LIST: A comma separated list of the host names of all LISTSERV Maestro servers of your LISTSERV Maestro installation. This may be one, two, or three names, depending on how many servers the LISTSERV Maestro components are distributed on. This list must not contain spaces, line breaks or anything but the host names and the separating commas. The host names must be the same host names that have been used for the respective server certificates. The name of the server on which this `ssl.ini` file is stored must be included as well.

KEYSTORE_FILE: The absolute path to the keystore file, including the drive letter, in which the signed server certificate for this server can be found. You cannot use a relative path name; the full path to the file is required. Remember to use either forward slashes ("/") or double backslashes ("\\") as the folder separator because INI files use the single backslash as an escape character.

PASSWORD: The password of the keystore file.

Security Issue: The password to the keystore and the certificate therein is included as plain text in this INI file. If unauthorized people have access to this file it can be a security breach. To protect your system, employ appropriate operating system or file system security measures so that only authorized people can access this file.

After you have created this file on all LISTSERV Maestro servers, restart LISTSERV Maestro (on all servers) to begin using secure inter-component communication.

Summary

Secure inter-component communication using SSL between the LISTSERV Maestro components is enabled if:

- Signed server certificates have been obtained and installed for all LISTSERV Maestro servers. If necessary, the trusted root certificates have been installed as well.
- The `ssl.ini` file has been created on all LISTSERV Maestro servers.
- LISTSERV Maestro has been restarted on all servers at least once since the steps above have been completed.
- If secure inter-component communication is enabled, it can be seen in the log-files. The log file of each component will contain a message with the startup messages similar to:

  ```
  XYZ: Initializing Secure Sockets Layer (SSL) for component
  communication
  ```

  Where "*XYZ*" is the name of the component.

References

LISTSERV Maestro Administrator's Manual

http://www.lsoft.com/resources/manuals.asp