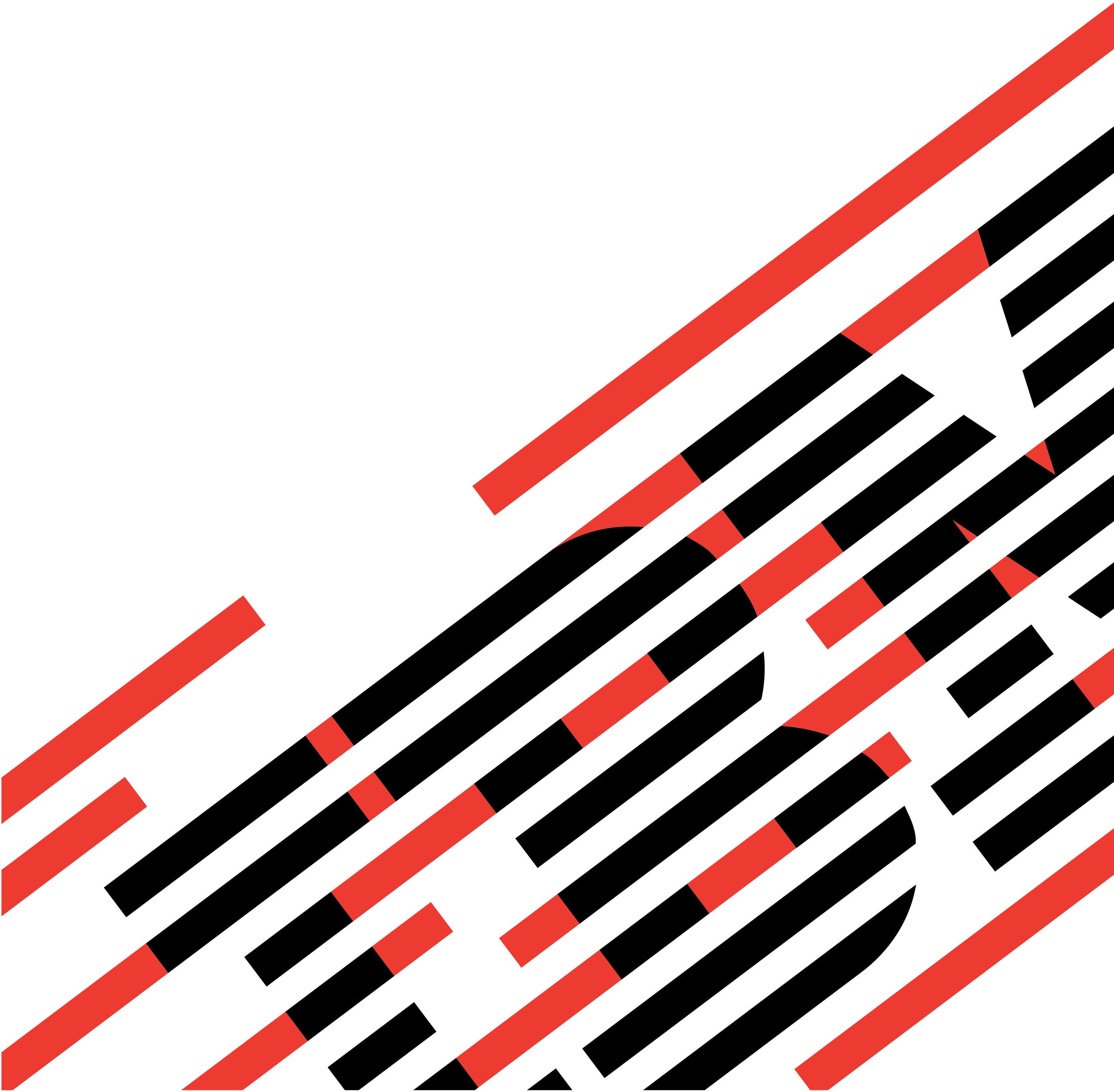
iSeries

Secure Sockets Layer (SSL)



iSeries

Secure Sockets Layer (SSL)

Contents

Part 1. Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) has become an industry standard for enabling applications for secure communication sessions over an unprotected network, such as the Internet. Use the following links to find more information about SSL and your iSeries™ server applications:

- **What's new for V5R2**
  makes note of new functions, or new information that is available to you, regarding SSL.
- **SSL scenarios**
  are a new addition to the SSL information, and are designed to increase your understanding of SSL on the iSeries server by providing possible examples of how SSL can work for you.
- **SSL concepts**
  includes supplemental information which provides some of the basic building blocks of the Secure Sockets Layer protocols.
- **Plan for SSL enablement**
  includes the prerequisites of SSL enablement on the iSeries server, as well as some helpful tips.
- **Secure applications with SSL**
  includes a list of applications that you can secure with SSL on the iSeries server.
- **Troubleshoot SSL**
  offers a basic guide for how to begin the procedure of troubleshooting SSL on the iSeries server.
- **Related information for SSL**
  includes links to additional information resources for your use.

Chapter 1. What’s new for V5R2

The 2058 Cryptographic Accelerator for iSeries is an available option at V5R2M0. This cryptographic hardware option is designed to improve the SSL performance of your iSeries server. See cryptographic hardware for more information on this option.

New Global Secure Kit (GSKit) application programming interface (API)

A new OS/400® Global Secure Toolkit (GSKit) API is available: `gsk_secure_soc_startInit()`. See Global Secure ToolKit (GSKit) APIs for more information.

To find other information about what’s new or changed this release, see the Memo to Users

How to see what’s new or changed

To help you see where technical changes have been made, this information uses:

- The

  »

  image to mark where new or changed information begins.
- The « image to mark where new or changed information ends.

Chapter 2. Print this topic

You can view or download the PDF version of this information. To do so, select Securing applications with SSL (about 215 KB or 34 pages).

Other information

You can also view or print any of the related information for this topic.

Saving PDF files

To save a PDF on your workstation for viewing or printing:
1. Right-click the PDF in your browser.
2. Click **Save Target As**.
3. Navigate to the directory in which you would like to save the PDF.
4. Click **Save**.

Downloading Adobe Acrobat Reader

If you need Adobe Acrobat Reader to view or print this information, you can download a copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Chapter 3. SSL scenarios

»

The following scenarios have been designed to help you maximize the benefits of enabling SSL on your iSeries server:

- Scenario: Secure Management Central with SSL
- Scenario: Secure FTP with SSL
- Scenario: Secure Telnet with SSL
- Scenario: Enhance iSeries SSL performance
- Scenario: Protect private keys with cryptographic hardware

«

SSL scenario: Secure Management Central with SSL

»

Situation

A company has just set up a wide area network (WAN) that includes several iSeries servers in remote locations (endpoint systems), that are centrally managed by one iSeries server located at the home office. This company's security specialist, Tom, uses the Management Central technology of his iSeries Navigator client to connect to his home office iSeries server (the central system). Tom wants to secure the connections between the central system and all endpoint servers with SSL.

Details

With the Management Central technology of iSeries Navigator, Tom can manage multiple systems through a single central system. By using SSL with Management Central, Tom can manage those systems **securely**. To use SSL with Management Central, Tom must secure iSeries Access for Windows® and iSeries Navigator on the PC from which he runs Management Central.

In a Management Central environment, Tom has two authentication levels:

Server authentication

Provides authentication of the endpoint system server certificate. The central system acts as an SSL client when connecting to an endpoint system. The endpoint system acts as an SSL server and must prove its identity by providing a certificate that was issued by a Certificate Authority that the central system trusts. There must be a valid certificate issued by a trusted CA for every endpoint system.

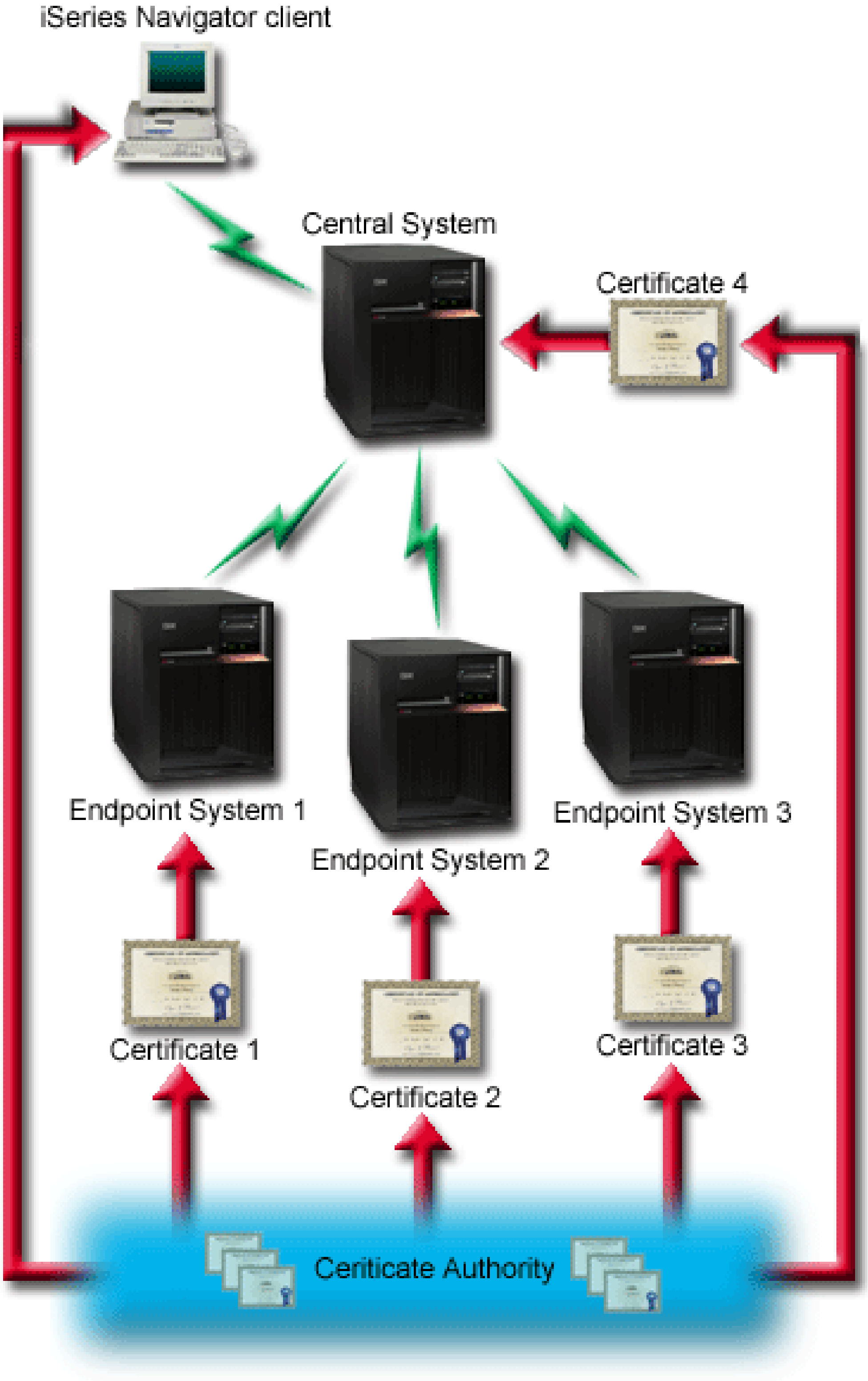
Client and server authentication

Provides authentication of both the central system and the endpoint system certificates. This is considered a stronger security level than the server authentication level. In other applications, this is known as client authentication, where the client must supply a valid trusted certificate. When the central system (SSL client) attempts to establish a connection with an endpoint system (SSL server), the central system and the endpoint system authenticate each other's certificates for certificate authority authenticity.

Unlike other applications, Management Central also provides authentication through a validation list, called Trusted Group validation list. Generally the validation list stores information that identifies the user, such as a user identification, and authentication information, such as password, personal identification number, or digital certificate. This authentication information is encrypted.

Most applications typically do not specify enabling both server and client authentication. This is because server authentication almost always occurs during an SSL session enablement. Many applications have client authentication configuration options. Management Central uses the term ″server and client authentication″ instead of client authentication because of the dual role that the central system plays in the network. When PC users connect to the central system and SSL is enabled, the central system acts as a server; however, when the central system is connecting to an endpoint system, it acts as a client. The following illustration shows how the central system operates as both a server and client in a network.

Note: In this illustration, the certificate associated with the Certificate Authority must be stored in the key database on the central system, and on all of the endpoint systems.



Prerequisites and assumptions

Tom must perform the following administration and configuration tasks (see the image, SSL-secured Management Central WAN), in order for SSL-enabled Management Central to work:

1. The iSeries server used with Management Central meets the prerequisites for SSL (see SSL Prerequisites).
2. The central system and all endpoint iSeries servers run V5R2 of OS/400. If they are at V5R1, install the following fixes (PTFs) for OS/400 (5722-SS1):
   a. SI01375
   b. SI01376
   c. SI01377
   d. SI01378
   e. SI01838
3. The iSeries Navigator PC client runs V5R2 of iSeries Access for Windows. If the client is at V5R1, install the service pack PTF SI01907 (or later) for V5R1 iSeries Access for Windows (5722-XE1). See the V5R1 Information Center, ″Securing Management Central″ page for more information.
4. Get a Certificate Authority (CA) for iSeries servers.
5. Creat a certificate, signed by the CA, for each iSeries server to be managed by the SSL-enabled Management Central Server.
6. Send the CA and a certificate to each iSeries server, and import them into the key database.
7. Assign the certificates with the Managment Central application identification, and the application identifications for all of the endpoint servers that iSeries Navigator uses:
   a. Start IBM® Digital Certificate Manager on the central server. If Tom needs to obtain or create certificates, or otherwise setup or change his certificate system, he does so now (see Using Digital Certificate Manager for information on setting up a certificate system).
   b. Click **Select a Certificate Store**.
   c. Select ***SYSTEM** and click **Continue**.
   d. Enter the *SYSTEM **Certificate Store password**, and click **Continue**. When the menu reloads, expand **Manage Applications**.
   e. Click **Update certificate assignment**.
   f. Select **Server** and click **Continue**.
   g. Select the **Management Central Server**, and click **Update certificate assignment**. This assigns a certificate to the Management Central server to use, in order to establish identity to iSeries Access for Windows clients.
   h. Click **Assign New Certificate**. DCM reloads to the **Update certificate assignment** page with a confirmation message.
   i. Click **Done**.
   j. Repeat this procedure for all endpoint servers that iSeries Navigator uses.
8. Set up iSeries Navigator:
   a. Selectively install the SSL component for iSeries Navigator.
   b. Download the CA from the system that it was created on.

      Note: If Tom chooses a certificate from a CA whose CA certificate is not in his iSeries Access for Windows client's key database, he needs to add the certificate to the database in order to use SSL.

Configuration steps

Before Tom can enable SSL on Management Central, he must install the prerequisite programs and set up digital certificates on the iSeries server (see the Prerequisites and assumptions for this scenario before continuing). Once he has met the prerequisites, he can complete the following procedures to enable SSL for Management Central.

Note: If SSL is enabled for iSeries Navigator, Tom must disable it before he can enable SSL for Management Central. If SSL is enabled for iSeries Navigator, and not Management Central, attempts by iSeries Navigator to connect with the Management Central central system will fail.

For server authentication (required):

1. Configure central system for server authentication
2. Configure endpoint systems for server authentication

For client authentication (optional):

Note: Client authentication configuration cannot be completed until server authentication is configured.

1. Configure central system for client authentication
2. Configure endpoint systems for client authentication

Configure central system for server authentication

SSL allows Tom to secure transmissions between a central system and an endpoint system, as well as between the iSeries Navigator client and the central system. SSL provides transport and authentication of certificates and encryption of data. An SSL-connection can only occur between an SSL-enabled central system and an SSL-enabled endpoint system. Tom must do the server authentication setup before he can do client authentication.

1. In iSeries Navigator, right-click **Management Central**, and select **Properties**.
2. Click the **Security** tab, and select **Use Secure Sockets Layer (SSL)**
3. Select **Server** for the authentication level.
4. Click **OK** to set this value on the central system.

   Note: Do **NOT** restart the Management Central Server until after the configuration of endpoint systems for server authentication is complete.
5. Configure endpoint systems for server authentication.

Configure endpoint systems for server authentication

After Tom has enabled SSL on the central system for server authentication, he needs to enable SSL for all endpoint systems for server authentication. To configure endpoint systems to use SSL and server authentication, he completes these tasks:

1. Expand the **Management Central** view.
2. **Compare and update system values for the endpoint systems:**
   a. Under **Endpoint Systems**, right-click on the central system and select **Inventory—>Collect**.
   b. Check the **System Values** option on the collect dialog, in order to collect the system values inventory for the central system. Uncheck any other options.
   c. Right-click **System Groups—>New System Group**.
   d. Define a new system group which includes all endpoint systems to connect to, using SSL.
   e. To display the new group, expand the list of system groups.
   f. After the collection is complete, right-click the new system group and select **System Values—>Compare and Update**.
   g. Verify the central system displays in the **Model system** field.

   h. Select the **Management Central** category and verify the following values, checking the box next to each:
      - Use Secure Sockets Layer is set to **Yes**.
      - The SSL authentication level is set to **Server**.

      These values are set on the central system during the procedure, Configure central system for server authentication.
   i. Click **OK** to set these values on the endpoint systems in the new system group.
   j. Wait for the **Compare and Update** process to complete before restarting the Management Central Server. This may take a few minutes.
3. **Restart the Management Central server on the central system:**
   a. In iSeries Navigator, expand **My Connections**.
   b. Expand the central system view.
   c. Expand **Network—> Servers** and select **TCP/IP**.
   d. Right-click **Management Central** and select **Stop**. The central sytem view collapses, and a message displays, explaining that you are no longer connected to the server.
   e. Once the Management Central server has stopped, click **Start** to restart it.
4. **Restart the Management Central server on all endpoint systems:**
   a. Expand the endpoint system that is being restarted.
   b. Expand **Network—> Servers** and select **TCP/IP**.
   c. Right-click **Management Central** and select **Stop**.
   d. Once the Management Central server has stopped, click **Start** to restart it.
   e. Repeat this procedure for each endpoint system.
5. **Activate SSL for the iSeries Navigator client:**
   a. In iSeries Navigator, expand **My Connections**.
   b. Right-click the central system, and select **Properties**.
   c. Click the **Secure Sockets** tab and select **Use Secure Sockets Layer (SSL) for connection**.
   d. Exit iSeries Navigator and restart it.

Now that Tom has completed the configuration for server authentication, he can perform the following optional client authentication procedures:
- Configure central system for client authentication
- Configure endpoint systems for client authentication

Client authentication provides validation of Certificate Authority and trusted group for both the endpoint systems and the central systems.

Configure the central system for client authentication

When the central system (SSL client) tries to use SSL to connect to an endpoint system (SSL server), the central system and the endpoint system authenticate each other's certificates through client authentication (called Certificate Authority and Trusted Group authentication in Management Central).
1. In iSeries Navigator, right-click **Management Central** and select **Properties**.
2. Click the **Security** tab and select **Use Secure Sockets Layer (SSL)**.
3. Select **Client and server** for the authentication level.
4. Click **OK** to set this value on the central system.

   Note: Do **NOT** restart the Management Central Server until all endpoint systems have been configured to use SSL with client and server authentication.
5. Configure endpoint systems for client authentication.

Configure endpoint systems for client authentication

1. **Compare and update system values for the endpoint systems:**

   Note: This task does not work for any endpoint iSeries servers that are running V4R5. See the V4R4 Redbook, ″Management Central: A Smart Way to Manage AS/400® Systems

   .″

   a. Under **Endpoint Systems**, right-click on the central system and select **Inventory—>Collect**.

   b. Check the **System Values** option on the collect dialog, in order to collect the system values inventory for the central system. Uncheck any other options.

   c. Right-click **System Groups—>New System Group**.

   d. Define a new system group that includes all the endpoint systems to connect to, using SSL.

   e. To display the new group, expand the list of system groups.

   f. After the collection is complete, right-click the new system group and select **System Values—>Compare and Update**.

   g. Verify that central system displays in the **Model System** field.

   h. Select the **Management Central** category and verify the following:

      - Use Secure Sockets Layer is set to **Yes**.
      - The SSL authentication level is set to **Client and Server**.

      These values are set on the central system during the procedure, Configure the central system for client authentication. Check the **Update** box next to each value.

   i. Click **OK** to set these values on the endpoint systems in the new system group.

2. **Copy the validation list to the endpoint systems:**

   a. In iSeries Navigator, expand **Management Central—>Definitions**.

   b. Right-click **Package**, and select **New Definition**.

   c. In the **New Definition** window, work with the following:

      - **Name:** Type the name of the definition.
      - **Source system:** Select the name of the central system.
      - **Selected files and folders:** Click in the field, and type `/QSYS.LIB/QUSRSYS.LIB/QYPSVLDL.VLDL`.

   d. Click the **Options** tab, and select **Replace existing file with the file being sent**.

   e. Click **Advanced**.

   f. In the **Advanced Options** window, specify **Yes** to allow object differences on restore.

   g. Click **OK** to refresh the list of definitions and display the new package.

   h. Right-click the new package, and select **Send**.

   i. In the **Send** dialog: Add the trusted group, remove any others, and click **OK**. The Trusted group is the system group you defined in Step 1 of this procedure.

      Note: The **Send** task will always fail on the central system, because it is always the source system. The **Send** task should complete successfully on all endpoint systems.

3. **Restart the Management Central server on the central system:**

   a. In iSeries Navigator, expand **My Connections**.

   b. Expand the central system.

   c. Expand **Network—> Servers** and select **TCP/IP**.

   d. Right-click **Management Central** and select **Stop**. The central sytem view collapses, and a message displays, explaining that you are no longer connected to the server.

   e. Once the Management Central server has stopped, click **Start** to restart it.

4. **Restart the Management Central server on all endpoint systems:**

   Note: Repeat this procedure for each endpoint system.

   a. Expand the endpoint system being restarted.
   b. Expand **Network—> Servers** and select **TCP/IP**.
   c. Right-click **Management Central** and select **Stop**.
   d. Once the Management Central server has stopped, click **Start** to restart it.

«

Chapter 4. SSL concepts

With the SSL protocol, you can establish secure connections between clients and server applications which provide authentication of one or both endpoints of the communication session. SSL also provides privacy and integrity of the data that client and server applications exchange.

The following conceptual information is provided to help you gain a better understanding of the relationship between SSL and the iSeries server:

- History of SSL
- How SSL works
- Supported SSL and Transport Layer Security (TLS) protocols
- Server authentication
- Client authentication

History of SSL

The Secure Sockets Layer Protocol (SSL) was developed by Netscape in 1994 as a response to the growing concern over security on the Internet. Although SSL was originally developed for securing web browser and server communications, the specification was designed in such a way that other applications, such as TELNET and FTP, could also be enabled to use SSL. See Supported SSL and Transport Layer Security (TLS) protocols for more information on SSL and related protocols.

How SSL works

SSL is actually two protocols. The protocols are the record protocol and the handshake protocol. The record protocol controls the flow of the data between the two endpoints of an SSL session.

The handshake protocol authenticates one or both endpoints of the SSL session and establishes a unique symmetric key used to generate keys to encrypt and decrypt data for that SSL session. SSL uses asymmetric cryptography, digital certificates, and SSL handshake flows to authenticate one or both endpoints of the SSL session. Usually, the server is authenticated and optionally the client is authenticated. A digital certificate, issued by a Certificate Authority, can be assigned to each of the endpoints or to the applications using SSL on each endpoint of the connection.

The digital certificate is comprised of a public key and some identifying information that has been digitally signed by a trusted Certificate Authority (CA). Each public key has an associated private key. The private key is not stored with or as part of the certificate. In both server and client authentication, the endpoint which is being authenticated must prove that it has access to the private key associated with the public key contained within the digital certificate.

SSL handshakes are performance intensive operations because of the cryptographic operations using the public and private keys. After an initial SSL session has been established between two endpoints, the SSL session information for these two endpoints and applications can be cached in secure memory to speed up subsequent SSL session enablements. When an SSL session is resumed, the two endpoints use an abbreviated handshake flow to authenticate that each has access to unique information without using the public or private keys. If both can prove that they have access to this unique information, then new symmetric keys are established and the SSL session is ″resumed″. For TLS Version 1.0 and SSL Version 3.0 sessions, cached information will not remain in the secure memory for greater than 24 hours. In V5R2M0, SSL handshake performance impacts on the main CPU can be minimized by using cryptographic hardware.

Supported SSL and Transport Layer Security (TLS) protocols

There are several versions of the SSL protocol defined. The latest version, the Transport Layer Security Protocol (TLS), is based on SSL 3.0 and is a product of the Internet Engineering Task Force (IETF). The OS/400 implementation supports the following versions of the SSL and TLS protocols:

- TLS Version 1.0
- TLS Version 1.0 with SSL Version 3.0 compatibility

  Notes:

  1. Specifying TLS Version 1.0 with SSL Version 3.0 compatibility means that TLS will be negotiated if possible and if that is not possible then SSL Version 3.0 will be negotiated. If SSL Version 3.0 cannot be negotiated then the SSL handshake will fail.
  2. We also support TLS Version 1.0 with SSL Version 3.0 and SSL Version 2.0 compatibility. This is specified with the protocol value of **ALL**, which means that TLS will be negotiated if possible and if that is not possible then SSL Version 3.0 will be negotiated. If SSL Version 3.0 cannot be negotiated, SSL Version 2.0 will be negotiated. If SSL Version 2.0 cannot be negotiated, then the SSL handshake will fail.
- SSL Version 3.0
- SSL Version 2.0
- SSL Version 3.0 with SSL Version 2.0 compatibility

SSL Version 3.0 versus SSL Version 2.0

SSL version 3.0 is an almost totally different protocol compared to SSL Version 2.0. Some of the major differences between the two protocols include:

- SSL Version 3.0 handshake protocol flows are different than SSL Version 2.0 handshake flows.
- SSL Version 3.0 uses the BSAFE 3.0 implementation from RSA Data Security, Inc. BSAFE 3.0 includes a number of timing attack fixes and the SHA-1 hashing algorithm. The SHA-1 hashing algorithm is considered to be more secure than the MD5 hashing algorithm. Having SHA-1 allows SSL Version 3.0 to support additional cipher suites which use SHA-1 instead of MD5.
- SSL Version 3.0 protocol reduces man-in-the-middle (MITM) type of attacks from occurring during SSL handshake processing. In SSL Version 2.0, it was possible, though unlikely, that a MITM attack could accomplish cipher specification weakening. Weakening the cipher could allow an unauthorized person to break the SSL session key.

TLS Version 1.0 versus SSL Version 3.0

Based on SSL Version 3.0, Transport Layer Security (TLS) Version 1.0 is the latest industry standard SSL protocol. Its specifications are defined by the Internet Engineering Task Force (IETF) in RFC 2246, ″The TLS Protocol.″

The major goal of TLS is to make SSL more secure and to make the specification of the protocol more precise and complete. TLS provides these enhancements over SSL Version 3.0:

- A more secure MAC algorithm
- More granular alerts
- Clearer definitions of ″gray area″ specifications

Any iSeries server applications that are enabled for SSL will automatically obtain TLS support unless the application has specifically requested to use only SSL Version 3.0 or SSL Version 2.0.

TLS provides the following security improvements:

- **Key-Hashing for Message Authentication**
  TLS uses Key-Hashing for Message Authentication Code (HMAC), which ensures that a record cannot be altered while travelling over an open network such as the Internet. SSL Version 3.0 also provides keyed message authentication, but HMAC is considered more secure than the (Message Authentication Code) MAC function that SSL Version 3.0 uses.
- **Enhanced Pseudorandom Function (PRF)**
  PRF is used for generating key data. In TLS, the PRF is defined with the HMAC. The PRF uses two hash algorithms in a way which guarantees its security. If either algorithm is exposed then the data will remain secure as long as the second algorithm is not exposed.
- **Improved finished message verification**
  Both TLS Version 1.0 and SSL Version 3.0 provide a finished message to both endpoints that authenticates that the exchanged messages were not altered. However, TLS bases this finished message on the PRF and HMAC values, which again is more secure than SSL Version 3.0.
- **Consistent certificate handling**
  Unlike SSL Version 3.0, TLS attempts specify the type of certificate which must be exchanged between TLS implementations.
- **Specific alert messages**
  TLS provides more specific and additional alerts to indicate problems that either session endpoint detects. TLS also documents when certain alerts should be sent.

Server authentication

With server authentication, the client will ensure that the server certificate is valid and that it is signed by a certificate authority (CA) which the client trusts. SSL will use asymmetric cryptography and handshake protocol flows to generate a symmetric key which will be used only for this unique SSL session. This key is used to generate a set of keys which are used for encrypting and decrypting data which will flow over the SSL session. Subsequently, when a SSL handshake has completed, one or both ends of the communication link will have been authenticated and a unique key will have been generated to encrypt and decrypt the data. Once the handshake is completed then application layer data will flow encrypted across that SSL session.

Client authentication

Many applications allow the option to enable client authentication. With client authentication, the server will ensure that the client certificate is valid and that it is signed by a Certificate Authority which the server trusts. The following iSeries server applications support client authentication:

- IBM HTTP Server (original)
- IBM HTTP Server (powered by Apache)
- FTP server
- Telnet server
- Management Central endpoint system
- Directory Services (LDAP)

Chapter 5. Plan for SSL enablement

When planning to enable SSL on an iSeries server, consider the following:

- SSL prerequisites
- What type of digital certificates you want, and where to obtain them

SSL Prerequisites:

- IBM Digital Certificate Manager (DCM), option 34 of OS/400 (5722-SS1)
- TCP/IP Connectivity Utilities for iSeries (5722-TC1)
- IBM HTTP Server for iSeries (5722-DG1)
- If you are trying to use the HTTP server to use the DCM, be sure you have the IBM Developer Kit for Java™ (5722–JV1) installed, or the HTTP admin server will not start.
- The IBM Cryptographic Access Provider product, 5722-AC3 (128-bit). The bit size for this product indicates the maximum size of the secret material within the symmetric keys that can be used in cryptographic operations. The size allowed for a symmetric key is controlled by the export and import laws of each country. A higher bit size results in a more secure connection.
- You may also want to install cryptographic hardware to use with SSL to speed up the SSL handshake processing. As of release V5R2M0, the following cryptographic hardware options are available to you, for use with your iSeries server:
  - 2058 Cryptographic Accelerator (Hardware Feature code 4805)
  - 4758 Cryptographic Coprocessor (Hardware Feature codes 4801 or 4802)

  If you want to install cryptographic hardware, you must also install Option 35, the Cryptographic Service Provider.

If you want to use SSL with any iSeries Access for Windows or IBM Toolbox for Java component you must also install the iSeries Client Encryption product, 5722-CE3 (128-bit). iSeries Access for Windows needs this product in order to establish the secure connection.

Note: You do not need to install a Client Encryption Product to use the PC5250 emulator that is shipped with the Personal Communications product. Personal Communications has its own built-in encryption code.

Digital certificates

See Using public certificates versus issuing private certificates to better understand the differences between public and private digital certificates, and your options for obtaining them.

IBM Digital Certificate Manager (DCM) is the iSeries server solution for managing digital certificates. To find out more about DCM, see the Information Center topic Using Digital Certificate Manager.

Chapter 6. Secure applications with SSL

You can secure the following iSeries server applications with SSL:

- IBM HTTP Server for iSeries (original)
- IBM HTTP Server for iSeries (powered by Apache)
- FTP server
- Telnet server
- Distributed relational database architecture (DRDA®) and distributed data management (DDM) server
- Management Central
- Directory Services Server (LDAP)
- Enterprise Identity Mapping (EIM)
- iSeries Access for Windows applications, including iSeries Navigator
- Applications that are written to the iSeries Access for Windows set of application programming interfaces (APIs)
- Programs developed with Developer Kit for Java and client applications that use IBM Toolbox for Java
- Applications developed using the secure sockets Application Programmable Interfaces (APIs) supported on the iSeries server. The supported APIs are Global Secure Toolkit (GSKit) and the SSL_ iSeries native APIs. See the Secure Sockets APIs for information on both GSKit and SSL_APIs.

Chapter 7. Troubleshoot SSL

»

This very basic troubleshooting information is intended to help you thin out the list of possible problems that the iSeries server can encounter with SSL. It is important to understand that this is not a comprehensive source for troubleshooting information, but simply a guide.

Verify that the following statements are true:

- You have met the prerequisites for SSL on the iSeries server (see SSL Prerequisites).
- If you are using the Management Central technology of iSeries Navigator with a V5R1 system, you have installed the following PTFs on your system:
  - si01375
  - si01376
  - si01377
  - si01378
  - si01838
- Your certificate authority and certificates are valid and have not expired.

If you have verified that the previous statements are true for your system, and you still have an SSL-related problem on the iSeries server, you can try the following options:

- The SSL error code in the server joblog can be cross referenced in an error table to find more information on the error. See the Secure socket API error code messages page to access information on secure socket error code messages. For example, this table maps the `-93`that might be seen in a server joblog to the constant `SSL_ERROR_SSL_NOT_AVAILABLE`.
  - A negative return code (indicated by the dash before the code number) indicates that you are using an SSL_ API.
  - A positive return code indicates that you are using a GSKit API. Programmers can code the `gsk_strerror()`or `SSL_Strerror()` API in their programs to obtain a brief description of an error return code. Some applications make use of this API and print out a message to the joblog containing this sentence.

  If more detailed information is required, the message id provided in the table can be displayed on an iSeries server to show potential cause and recovery for this error. Additional documentation explaining these error codes may be located in the individual secure socket API that has returned the error.
- The following two header files contain the same constant names for System SSL return codes as the table, but without the message ID cross reference:
  - `QSYSINC/H.GSKSSL`
  - `QSYSINC/H.SSL`

  Remember that although the names of the System SSL return codes remain constant in these two files, more than one unique error can be associated with each return code.

For more troubleshooting information regarding the iSeries server, see the Troubleshooting and service page.«

Chapter 8. Related information

»

You can find additional SSL information in the following sources:

IBM Sources

- The SSL and Java Secure Socket Extension (JSSE) page includes a brief description of JSSE and how you can use it.
- The Java Secure Socket Layer (JSSL) page includes a brief description of JSSL and how you can use it.
- The IBM Toolbox for Java page includes a brief description of the Java classes available, and how you can use them.

Request for comments

- RFC 2246: ″The TLS Protocol Version 1.0 ″ explains the TLS protocol in detail.
- RFC2818: ″HTTP Over TLS″ describes how to use TLS to secure HTTP connections over the Internet.

Other sources

- The The SSL Protocol Version 3.0 document explains SSL Protocol Version 3.0 in great detail.

«

Printed in U.S.A.