New CICS support for Secure Sockets Layer

Peter Havercan, Senior CICS Developer

CICS® Transaction Server has had support for Secure Sockets Layer (SSL) since Version 1 Release 3, but the support has been considerably enhanced in Version 3 Release 1. Many of the enhancements arise from exploitation of changes in the underlying z/OS® support for SSL, in the System SSL component, but others arise from a restructure of the CICS code itself.

SSL is a security protocol that is used extensively on the Internet to allow private secure communications over the public network. SSL was defined by the Netscape Corporation and currently exists as Versions 2 and 3. (Version 1 was never publicly available.) It has now been superseded by Transport Layer Security (TLS), which is very similar to SSLv3, but is sufficiently different that it cannot interoperate with it. CICS now supports both TLS and SSLv3.

An SSL connection has two phases: a handshake phase, when a client and server attempt to negotiate an encryption algorithm; and a payload phase, when they use that algorithm to exchange encrypted data. The negotiated algorithm is actually specified as a cipher suite. In addition to specifying an encryption algorithm and an associated keylength, the cipher suite also specifies a message authentication code (MAC) algorithm, which is used to add an authentication code to each exchanged message. The MAC confirms that messages have not been corrupted (either accidentally on deliberately) during transmission.

System SSL is a component of MVS™ that was introduced in OS/390® Version 2 Release 7, and CICS was early to exploit this in Transaction Server Version 1 Release 3. At that time, the export of software containing strong encryption was tightly controlled, so CICS introduced a system initialization parameter to specify what level of encryption was installed in the underlying system. This ENCRYPTION parameter was used to specify the set of cipher suites that could be negotiated between CICS and its partner system during the SSL handshake. Apart from its ability to specify a maximum encryption ability, CICS did not allow the system programmer to specify the cipher suites in any other way.

Selectable cipher suites

CICS has now introduced a much more granular control of the cipher suites that can be negotiated with an SSL partner. In each of the CICS resources that define communication over sockets, a new CIPHERS parameter has been introduced. This can be used to specify precisely the list of cipher suites that are acceptable: if the SSL partner cannot or will not use any of the specified cipher codes, the SSL connection is not established. The CICS resources on which CIPHERS can be specified are TCPIPSERVICE, for inbound sockets connection, CORBASERVER, for outbound IIOP sockets connection, and the new URIMAP resource for outbound HTTP connections. CIPHERS can also be specified on the new EXEC CICS WEB OPEN command, which is also used to define outbound HTTP socket connections.

The CIPHERS list is specified as a sequence of two-digit hexadecimal codes. The values of the hexadecimal codes are listed in the formal specifications for SSL and TLS. The cipher suite codes are also known in these specifications by formal names, but CICS does not use these names, although it does interpret the hexadecimal codes into their formal names in some of its trace records. For example, hexadecimal cipher suite code 0A is known formally as SSL_RSA_WITH_3DES_EDE_CBC_SHA, which means that it uses RSA public key encryption to negotiate triple-DES (168-bit) key and uses SHA as a MAC algorithm. The power of being able to specify precisely which ciphers are used in the negotiation means that you can be sure that the SSL partner cannot "negotiate down" to a weaker cipher than you expect, which was possible in earlier releases of CICS. A typical ciphers specification that includes only very strong cipher specifications would be CIPHERS(352F0A0504).

Two new cipher suite codes that are available in CICS TS 3.1 are 2F and 35, which specify the advanced encryption standard (AES) encryption algorithm with 128- and 256-bit keysizes, respectively. These algorithms are implemented by encryption hardware on zSeries® machines, so they provide a high degree of security with quite high performance. The cipher codes supported by CICS are exactly those supported by the underlying z/OS operating system. On z/OS 1.6, a whole set of ciphers using the Diffie-Hellman encryption scheme became available.

During initial development of CICS TS 3.1, support was still available for SSLv2, but a discovery was made. When CICS was defined with only the AES ciphers, and was accessed by a browser that did not support them but continued to support SSLv2, the browser would negotiate down to an SSLv2 connection. This was the exact opposite of the high-security connection that was intended. Accordingly, SSLv2 support has now been removed from CICS TS 3.1.

Certificate Revocation Lists

During an SSL handshake there is an exchange of certificates. The server side of the connection always sends a certificate to the client and, at the server's request, the client may be required to send a certificate to the server. During the handshake, each side of the conversation may check whether the certificates are revoked. A certificate is revoked when it is included in a certificate revocation list, which is a list of certificates that have been revoked by the certificate authority that issued them. Typically, certificate authorities who issue the certificates – such as Verisign, Thawte, Geotrust and Equifax – also issue corresponding certificate revocation lists on their Web sites.

Users of certificates – SSL clients and servers – should download certificate revocation lists (CRLs) from the Web sites of the certificate issuers. However, it is impractical to download a CRL every time a certificate is to be validated so, in practice, the CRLs are downloaded occasionally and saved locally in the user's system. System SSL assumes that the CRLs have been stored locally in an LDAP server, but it does not provide a way of loading the CRLs into LDAP. Therefore, CICS offers the CCRL utility transaction to download CRLs from a Web site and save them on an LDAP server. For security reasons, the LDAP server has to be identified to CICS using a profile in the security manager, which is then identified by the CRLPROFILE system initialization parameter. The same LDAP server that is used by the CCRL transaction is then also used by the CICS SSL support to validate certificates. If a certificate used in the SSL handshake is found in a revocation list in the LDAP server, the certificate is rejected and the connection is not established.

Sysplex-wide session ID caching

Because an SSL handshake is a relatively expensive operation, the SSL protocol contains an optional optimization. During the handshake, the client and server negotiate a 32-byte session identifier. If the connection is closed and later reestablished, the client presents that same session ID to the server again. The server may recognize the session ID and reestablish the secure connection without a full expensive handshake. In order to allow this, the server has to keep a cache of issued session IDs. Session IDs are only maintained in the cache for a limited time to prevent possible replays and reconnection by an unauthorized client.

A possible problem arises if multiple SSL servers are deployed using the same IP address. This is a common configuration used for load balancing. If a previously negotiated session ID is cached by one server, but the client reconnects to a different server at the same IP address, the second server will have no copy of the session ID in its cache and will force a full handshake instead of the optimized one. CICS can now be configured to use sysplex-wide caching of session IDs. This is specified using the SSLCACHE=SYSPLEX system initialization parameter. When this option is chosen, all the session IDs for all related servers are shared, so a connection established by one server can be reestablished by another one without needing a full handshake.

The shared cache is managed by System SSL in an address space created by a separate MVS started task named GSKSRVR. If you need to use the shared cache, you have to be sure that this started task is active. Although the sysplex-wide caching option may improve performance if an IP address is actually being shared between servers, it should not be used when each server has its own IP address, since sharing the session ID cache does introduce extra overhead.
Increasing the number of SSL connections

Earlier releases of CICS were very constrained in the number of simultaneous connections that could be established within one CICS region. The system SSL service has to be invoked in the C language from a POSIX-enabled Language Environment enclave. The natural way of providing multithreading in such an environment is to use pthreads, but CICS had never been able to support pthreads on the TCBs that it manages. This meant that, in earlier releases, CICS had to implement multithreading of SSL connections by assigning a separate LE enclave to each connection, and assigning each enclave to a single TCB. The storage costs of this were huge: each enclave consumed about 20K of storage below the 16M line, so it was only practical to allocate about 300 simultaneous SSL connections. Even this was only possible by tweaking the DSALIM system initialization parameter and eliminating all other uses of below-the-line storage.

In CICS TS 3.1 this constraint has been finally eliminated. The CICS dispatcher has now been changed to understand pthreads and to use them as CICS-managed TCBs in the Open Transaction Environment. (This is only available for internal CICS processes like SSL, though. It is not available for application programs.) As a result of the pthread support, CICS only needs to allocate a single Language Environment enclave for all the SSL connections, and allows vastly more simultaneous connections to take place using pthreads within that enclave. Instead of only 300 simultaneous SSL connections into CICS, it is now possible to deploy tens of thousands of connections.

Conclusion

Secure Sockets support in CICS TS 3.1 is considerably improved over previous CICS releases. The TLS protocol and the most modern AES encryption algorithms are supported. Selection of cipher lists allows you to select the precise level of encryption that CICS negotiates with its partners. Sysplex-wide session ID caching can improve performance when cloned regions share the same IP address. Certificate revocation lists can be saved and used in an LDAP server. And, most significantly, the number of simultaneous connections has been increased a hundred-fold.

Author's biography

The author of this paper is Peter Havercan, who is a senior CICS developer at IBM's Hursley Laboratory. Peter has twenty years of experience in designing and developing mainframe CICS software, especially in the areas of CICS security and CICS Web support. As a former MVS system programmer, he retains an insight into the internals of z/OS, which he uses to advantage when developing CICS. Peter is also a frequent presesenter at CICS technical conferences.