NAME

ssl - OpenSSL SSL/TLS library

SYNOPSIS

See the individual manual pages for details.

DESCRIPTION

The OpenSSL **ssl** library implements several versions of the Secure Sockets Layer, Transport Layer Security, and Datagram Transport Layer Security protocols. This page gives a brief overview of the extensive API and data types provided by the library.

An **SSL_CTX** object is created as a framework to establish TLS/SSL enabled connections (see **SSL_CTX_new**(3)). Various options regarding certificates, algorithms etc. can be set in this object.

When a network connection has been created, it can be assigned to an **SSL** object. After the **SSL** object has been created using **SSL_new**(3), **SSL_set_fd**(3) or **SSL_set_bio**(3) can be used to associate the network connection with the object.

When the TLS/SSL handshake is performed using **SSL_accept**(3) or **SSL_connect**(3) respectively. **SSL_read_ex**(3), **SSL_read**(3), **SSL_write_ex**(3) and **SSL_write**(3) are used to read and write data on the TLS/SSL connection. **SSL_shutdown**(3) can be used to shut down the TLS/SSL connection.

DATA STRUCTURES

Here are some of the main data structures in the library.

SSL_METHOD (SSL Method)

This is a dispatch structure describing the internal **ssl** library methods/functions which implement the various protocol versions (SSLv3 TLSv1, ...). It's needed to create an **SSL_CTX**.

SSL_CIPHER (SSL Cipher)

This structure holds the algorithm information for a particular cipher which are a core part of the SSL/TLS protocol. The available ciphers are configured on a **SSL_CTX** basis and the actual ones used are then part of the **SSL_SESSION**.

SSL_CTX (SSL Context)

This is the global context structure which is created by a server or client once per program lifetime and which holds mainly default values for the **SSL** structures which are later created for the connections.

SSL_SESSION (SSL Session)

This is a structure containing the current TLS/SSL session details for a connection: **SSL_CIPHER**s, client and server certificates, keys, etc.

SSL (SSL Connection)

This is the main SSL/TLS structure which is created by a server or client per established connection. This actually is the core structure in the SSL API. At run-time the application usually deals with this structure which has links to mostly all other structures.

HEADER FILES

Currently the OpenSSL **ssl** library provides the following C header files containing the prototypes for the data structures and functions:

<openssl/ssl.h>

This is the common header file for the SSL/TLS API. Include it into your program to make the API of the **ssl** library available. It internally includes both more private SSL headers and headers from the **crypto** library. Whenever you need hard-core details on the internals of the SSL API, look inside this header file. This file also includes the others listed below.

<openssl/ssl2.h>

Unused. Present for backwards compatibility only.

<openssl/ssl3.h>

This is the sub header file dealing with the SSLv3 protocol only.

<openssl/tls1.h>

This is the sub header file dealing with the TLSv1 protocol only.

COPYRIGHT