NAME

SSL_COMP_add_compression_method, SSL_COMP_get_compression_methods, SSL_COMP_get0_name, SSL_COMP_get_id, SSL_COMP_free_compression_methods - handle SSL/TLS integrated compression methods

SYNOPSIS

```
#include <openssl/ssl.h>

int SSL_COMP_add_compression_method(int id, COMP_METHOD *cm);
STACK_OF(SSL_COMP) *SSL_COMP_get_compression_methods(void);
const char *SSL_COMP_get0_name(const SSL_COMP *comp);
int SSL_COMP_get_id(const SSL_COMP *comp);
```

Deprecated:

```
#if OPENSSL_API_COMPAT < 0x10100000L
void SSL_COMP_free_compression_methods(void)
#endif
```

DESCRIPTION

SSL_COMP_add_compression_method() adds the compression method **cm** with the identifier **id** to the list of available compression methods. This list is globally maintained for all SSL operations within this application. It cannot be set for specific SSL_CTX or SSL objects.

SSL_COMP_get_compression_methods() returns a stack of all of the available compression methods or NULL on error.

SSL_COMP_get0_name() returns the name of the compression method **comp**.

SSL_COMP_get_id() returns the id of the compression method **comp**.

In versions of OpenSSL prior to 1.1.0 *SSL_COMP_free_compression_methods()* freed the internal table of compression methods that were built internally, and possibly augmented by adding *SSL_COMP_add_compression_method()*. However this is now unnecessary from version 1.1.0. No explicit initialisation or de-initialisation is necessary. See *OPENSSL_init_crypto(3)* and *OPENSSL_init_ssl(3)*. From OpenSSL 1.1.0 calling this function does nothing.

NOTES

The TLS standard (or SSLv3) allows the integration of compression methods into the communication. The TLS RFC does however not specify compression methods or their corresponding identifiers, so there is currently no compatible way to integrate compression with unknown peers. It is therefore currently not recommended to integrate compression into applications. Applications for non-public use may agree on certain compression methods. Using different compression methods with the same identifier will lead to connection failure.

An OpenSSL client speaking a protocol that allows compression (SSLv3, TLSv1) will unconditionally send the list of all compression methods enabled with *SSL_COMP_add_compression_method()* to the server during the handshake. Unlike the mechanisms to set a cipher list, there is no method available to restrict the list of compression method on a per connection basis.

An OpenSSL server will match the identifiers listed by a client against its own compression methods and will unconditionally activate compression when a matching identifier is found. There is no way to restrict the list of compression methods supported on a per connection basis.

If enabled during compilation, the OpenSSL library will have the *COMP_zlib()* compression method available.

WARNINGS

Once the identities of the compression methods for the TLS protocol have been standardized, the compression API will most likely be changed. Using it in the current state is not recommended.

RETURN VALUES

SSL_COMP_add_compression_method() may return the following values:

0 The operation succeeded.

1 The operation failed. Check the error queue to find out the reason.

SSL_COMP_get_compression_methods() returns the stack of compressions methods or NULL on error.

SSL_COMP_get0_name() returns the name of the compression method or NULL on error.

SSL_COMP_get_id() returns the name of the compression method or -1 on error.

SEE ALSO

ssl(3)

HISTORY

SSL_COMP_free_compression_methods() was deprecated in OpenSSL 1.1.0. *SSL_COMP_g et0_name()* and *SSL_comp_get_id()* were added in OpenSSL 1.1.0d.

COPYRIGHT