NAME

SSL_CTX_set_msg_callback, SSL_CTX_set_msg_callback_arg, SSL_set_msg_callback, SSL_set_msg_callback_arg - install callback for observing protocol messages

SYNOPSIS

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

void SSL_CTX_set_msg_callback(SSL_CTX *ctx, void (*cb)(int write_p, int version,
void SSL_CTX_set_msg_callback_arg(SSL_CTX *ctx, void *arg);

void SSL_set_msg_callback(SSL *ssl, void (*cb)(int write_p, int version, int con
void SSL_set_msg_callback_arg(SSL *ssl, void *arg);
```

DESCRIPTION

SSL_CTX_set_msg_callback() or *SSL_set_msg_callback()* can be used to define a message callback function *cb* for observing all SSL/TLS protocol messages (such as handshake messages) that are received or sent. *SSL_CTX_set_msg_callback_arg()* and *SSL_set_msg_callback_arg()* can be used to set argument *arg* to the callback function, which is available for arbitrary application use.

SSL_CTX_set_msg_callback() and *SSL_CTX_set_msg_callback_arg()* specify default settings that will be copied to new **SSL** objects by *SSL_new(3)*. *SSL_set_msg_callback()* and *SSL_set_msg_callback_arg()* modify the actual settings of an **SSL** object. Using a **0** pointer for *cb* disables the message callback.

When *cb* is called by the SSL/TLS library for a protocol message, the function arguments have the following meaning:

write_p
: This flag is **0** when a protocol message has been received and **1** when a protocol message has been sent.

version
: The protocol version according to which the protocol message is interpreted by the library. Currently, this is one of **SSL2_VERSION**, **SSL3_VERSION** and **TLS1_VERSION** (for SSL 2.0, SSL 3.0 and TLS 1.0, respectively).

content_type
: In the case of SSL 2.0, this is always **0**. In the case of SSL 3.0 or TLS 1.0, this is one of the **ContentType** values defined in the protocol specification (**change_cipher_spec(20)**, **alert(21)**, **handshake(22)**; but never **application_data(23)** because the callback will only be called for protocol messages).

buf, len
: *buf* points to a buffer containing the protocol message, which consists of *len* bytes. The buffer is no longer valid after the callback function has returned.

ssl
: The **SSL** object that received or sent the message.

arg
: The user-defined argument optionally defined by *SSL_CTX_set_msg_callback_arg()* or *SSL_set_msg_callback_arg()*.

NOTES

Protocol messages are passed to the callback function after decryption and fragment collection where applicable. (Thus record boundaries are not visible.)

If processing a received protocol message results in an error, the callback function may not be called. For example, the callback function will never see messages that are considered too large to be processed.

Due to automatic protocol version negotiation, *version* is not necessarily the protocol version used by the sender of the message: If a TLS 1.0 ClientHello message is received by an SSL 3.0-only server, *version* will be **SSL3_VERSION**.

SEE ALSO

ssl(3) , *SSL_new(3)*

COPYRIGHT