NAME

SSL_CTX_set_record_padding_callback, SSL_set_record_padding_callback, SSL_CTX_set_record_padding_callback_arg, SSL_set_record_padding_callback_arg, SSL_CTX_get_record_padding_callback_arg, SSL_get_record_padding_callback_arg, SSL_CTX_set_block_padding, SSL_set_block_padding - install callback to specify TLS 1.3 record padding

SYNOPSIS

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

void SSL_CTX_set_record_padding_callback(SSL_CTX *ctx, size_t (*cb)(SSL *s, int type, size_t len, void *arg));
int SSL_set_record_padding_callback(SSL *ssl, size_t (*cb)(SSL *s, int type, size_t len, void *arg));

void SSL_CTX_set_record_padding_callback_arg(SSL_CTX *ctx, void *arg);
void *SSL_CTX_get_record_padding_callback_arg(const SSL_CTX *ctx);

void SSL_set_record_padding_callback_arg(SSL *ssl, void *arg);
void *SSL_get_record_padding_callback_arg(const SSL *ssl);

int SSL_CTX_set_block_padding(SSL_CTX *ctx, size_t block_size);
int SSL_set_block_padding(SSL *ssl, size_t block_size);
```

DESCRIPTION

SSL_CTX_set_record_padding_callback() or **SSL_set_record_padding_callback()** can be used to assign a callback function *cb* to specify the padding for TLS 1.3 records. The value set in **ctx** is copied to a new SSL by **SSL_new()**. Kernel TLS is not possible if the record padding callback is set, and the callback function cannot be set if Kernel TLS is already configured for the current SSL object.

SSL_CTX_set_record_padding_callback_arg() and **SSL_set_record_padding_callback_arg()** assign a value **arg** that is passed to the callback when it is invoked. The value set in **ctx** is copied to a new SSL by **SSL_new()**.

SSL_CTX_get_record_padding_callback_arg() and **SSL_get_record_padding_callback_arg()** retrieve the **arg** value that is passed to the callback.

SSL_CTX_set_block_padding() and **SSL_set_block_padding()** pads the record to a multiple of the **block_size**. A **block_size** of 0 or 1 disables block padding. The limit of **block_size** is SSL3_RT_MAX_PLAIN_LENGTH.

The callback is invoked for every record before encryption. The **type** parameter is the TLS record type

that is being processed; may be one of SSL3_RT_APPLICATION_DATA, SSL3_RT_HANDSHAKE, or SSL3_RT_ALERT. The **len** parameter is the current plaintext length of the record before encryption. The **arg** parameter is the value set via **SSL_CTX_set_record_padding_callback_arg()** or **SSL_set_record_padding_callback_arg()**.

RETURN VALUES

The **SSL_CTX_get_record_padding_callback_arg()** and **SSL_get_record_padding_callback_arg()** functions return the **arg** value assigned in the corresponding set functions.

The **SSL_CTX_set_block_padding()** and **SSL_set_block_padding()** functions return 1 on success or 0 if **block_size** is too large.

The **cb** returns the number of padding bytes to add to the record. A return of 0 indicates no padding will be added. A return value that causes the record to exceed the maximum record size (SSL3_RT_MAX_PLAIN_LENGTH) will pad out to the maximum record size.

The **SSL_CTX_get_record_padding_callback_arg()** function returns 1 on success or 0 if the callback function is not set because Kernel TLS is configured for the SSL object.

NOTES

The default behavior is to add no padding to the record.

A user-supplied padding callback function will override the behavior set by **SSL_set_block_padding()** or **SSL_CTX_set_block_padding()**. Setting the user-supplied callback to NULL will restore the configured block padding behavior.

These functions only apply to TLS 1.3 records being written.

Padding bytes are not added in constant-time.

SEE ALSO

ssl(7), **SSL_new**(3)

HISTORY

The record padding API was added for TLS 1.3 support in OpenSSL 1.1.1.

The return type of **SSL_CTX_set_record_padding_callback()** function was changed to int in OpenSSL 3.0.

COPYRIGHT