



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'SSL\_CTX\_set\_cookie\_verify\_cb.3ossl' command**

```
$ man SSL_CTX_set_cookie_verify_cb.3ossl
```

```
SSL_CTX_SET_STATELESS_COOKIE_GESSL_CTX_SET_STATELESS_COOKIE_GENERATE_CB(3ossl)
```

### NAME

```
SSL_CTX_set_stateless_cookie_generate_cb,  
SSL_CTX_set_stateless_cookie_verify_cb, SSL_CTX_set_cookie_generate_cb,  
SSL_CTX_set_cookie_verify_cb - Callback functions for stateless TLS1.3  
cookies
```

### SYNOPSIS

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

```
void SSL_CTX_set_stateless_cookie_generate_cb(  
    SSL_CTX *ctx,  
    int (*gen_stateless_cookie_cb) (SSL *ssl,  
        unsigned char *cookie,  
        size_t *cookie_len));
```

```
void SSL_CTX_set_stateless_cookie_verify_cb(  
    SSL_CTX *ctx,  
    int (*verify_stateless_cookie_cb) (SSL *ssl,  
        const unsigned char *cookie,  
        size_t cookie_len));
```

```
void SSL_CTX_set_cookie_generate_cb(SSL_CTX *ctx,
```

```

        int (*app_gen_cookie_cb) (SSL *ssl,
                                unsigned char
                                *cookie,
                                unsigned int
                                *cookie_len));

void SSL_CTX_set_cookie_verify_cb(SSL_CTX *ctx,
                                int (*app_verify_cookie_cb) (SSL *ssl,
                                                            const unsigned
                                                            char *cookie,
                                                            unsigned int
                                                            cookie_len));

```

## DESCRIPTION

SSL\_CTX\_set\_stateless\_cookie\_generate\_cb() sets the callback used by SSL\_stateless(3) to generate the application-controlled portion of the cookie provided to clients in the HelloRetryRequest transmitted as a response to a ClientHello with a missing or invalid cookie.

gen\_stateless\_cookie\_cb() must write at most SSL\_COOKIE\_LENGTH bytes into cookie, and must write the number of bytes written to cookie\_len.

If a cookie cannot be generated, a zero return value can be used to abort the handshake.

SSL\_CTX\_set\_stateless\_cookie\_verify\_cb() sets the callback used by SSL\_stateless(3) to determine whether the application-controlled portion of a ClientHello cookie is valid. The cookie data is pointed to by cookie and is of length cookie\_len. A nonzero return value from verify\_stateless\_cookie\_cb() communicates that the cookie is valid. The integrity of the entire cookie, including the application-controlled portion, is automatically verified by HMAC before verify\_stateless\_cookie\_cb() is called.

SSL\_CTX\_set\_cookie\_generate\_cb() sets the callback used by DTLSv1\_listen(3) to generate the cookie provided to clients in the

HelloVerifyRequest transmitted as a response to a ClientHello with a missing or invalid cookie. `app_gen_cookie_cb()` must write at most `DTLS1_COOKIE_LENGTH` bytes into `cookie`, and must write the number of bytes written to `cookie_len`. If a cookie cannot be generated, a zero return value can be used to abort the handshake.

`SSL_CTX_set_cookie_verify_cb()` sets the callback used by `DTLSv1_listen(3)` to determine whether the cookie in a ClientHello is valid. The cookie data is pointed to by `cookie` and is of length `cookie_len`. A nonzero return value from `app_verify_cookie_cb()` communicates that the cookie is valid. The integrity of the cookie is not verified by OpenSSL. This is an application responsibility.

## RETURN VALUES

Neither function returns a value.

## SEE ALSO

`ssl(7)`, `SSL_stateless(3)`, `DTLSv1_listen(3)`

## HISTORY

`SSL_CTX_set_stateless_cookie_generate_cb()` and `SSL_CTX_set_stateless_cookie_verify_cb()` were added in OpenSSL 1.1.1.

## COPYRIGHT

Copyright 2018 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file `LICENSE` in the source distribution or at <https://www.openssl.org/source/license.html>.