NAME

SSL_set_num_tickets, SSL_get_num_tickets, SSL_CTX_set_num_tickets, SSL_CTX_get_num_tickets, SSL_new_session_ticket - control the number of TLSv1.3 session tickets that are issued

SYNOPSIS

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

int SSL_set_num_tickets(SSL *s, size_t num_tickets);
size_t SSL_get_num_tickets(const SSL *s);
int SSL_CTX_set_num_tickets(SSL_CTX *ctx, size_t num_tickets);
size_t SSL_CTX_get_num_tickets(const SSL_CTX *ctx);
int SSL_new_session_ticket(SSL *s);
```

DESCRIPTION

SSL_CTX_set_num_tickets() and **SSL_set_num_tickets()** can be called for a server application and set the number of TLSv1.3 session tickets that will be sent to the client after a full handshake. Set the desired value (which could be 0) in the **num_tickets** argument. Typically these functions should be called before the start of the handshake.

The default number of tickets is 2. Following a resumption the number of tickets issued will never be more than 1 regardless of the value set via **SSL_set_num_tickets()** or **SSL_CTX_set_num_tickets()**. If **num_tickets** is set to 0 then no tickets will be issued for either a normal connection or a resumption.

Tickets are also issued on receipt of a post-handshake certificate from the client following a request by the server using **SSL_verify_client_post_handshake**(3). These new tickets will be associated with the updated client identity (i.e. including their certificate and verification status). The number of tickets issued will normally be the same as was used for the initial handshake. If the initial handshake was a full handshake then **SSL_set_num_tickets()** can be called again prior to calling **SSL_verify_client_post_handshake()** to update the number of tickets that will be sent.

To issue tickets after other events (such as application-layer changes), **SSL_new_session_ticket()** is used by a server application to request that a new ticket be sent when it is safe to do so. New tickets are only allowed to be sent in this manner after the initial handshake has completed, and only for TLS 1.3 connections. By default, the ticket generation and transmission are delayed until the server is starting a new write operation, so that it is bundled with other application data being written and properly aligned to a record boundary. If the connection was at a record boundary when **SSL_new_session_ticket()** was called, the ticket can be sent immediately (without waiting for the next application write) by calling **SSL_do_handshake()**. **SSL_new_session_ticket()** can be called more than once to request additional tickets be sent; all such requests are queued and written together when it is

safe to do so and triggered by **SSL_write()** or **SSL_do_handshake()**. Note that a successful return from **SSL_new_session_ticket()** indicates only that the request to send a ticket was processed, not that the ticket itself was sent. To be notified when the ticket itself is sent, a new-session callback can be registered with **SSL_CTX_sess_set_new_cb**(3) that will be invoked as the ticket or tickets are generated.

SSL_CTX_get_num_tickets() and **SSL_get_num_tickets()** return the number of tickets set by a previous call to **SSL_CTX_set_num_tickets()** or **SSL_set_num_tickets()**, or 2 if no such call has been made.

RETURN VALUES

SSL_CTX_set_num_tickets(), **SSL_set_num_tickets()**, and **SSL_new_session_ticket()** return 1 on success or 0 on failure.

SSL_CTX_get_num_tickets() and **SSL_get_num_tickets()** return the number of tickets that have been previously set.

SEE ALSO

ssl(7)

HISTORY

SSL_new_session_ticket() was added in OpenSSL 3.0.0. **SSL_set_num_tickets()**, **SSL_get_num_tickets()**, **SSL_CTX_set_num_tickets()**, and **SSL_CTX_get_num_tickets()** were added in OpenSSL 1.1.1.

COPYRIGHT