NAME

RAND_add, RAND_seed, RAND_status, RAND_event, RAND_screen - add entropy to the PRNG

SYNOPSIS

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

void RAND_seed(const void *buf, int num);

void RAND_add(const void *buf, int num, double entropy);

int RAND_status(void);

#if OPENSSL_API_COMPAT < 0x10100000L
int RAND_event(UINT iMsg, WPARAM wParam, LPARAM lParam);
void RAND_screen(void);
#endif
```

DESCRIPTION

RAND_add() mixes the **num** bytes at **buf** into the PRNG state. Thus, if the data at **buf** are unpredictable to an adversary, this increases the uncertainty about the state and makes the PRNG output less predictable. Suitable input comes from user interaction (random key presses, mouse movements) and certain hardware events. The **entropy** argument is (the lower bound of) an estimate of how much randomness is contained in **buf**, measured in bytes. Details about sources of randomness and how to estimate their entropy can be found in the literature, e.g. RFC 1750.

RAND_add() may be called with sensitive data such as user entered passwords. The seed values cannot be recovered from the PRNG output.

OpenSSL makes sure that the PRNG state is unique for each thread. On systems that provide `/dev/urandom`, the randomness device is used to seed the PRNG transparently. However, on all other systems, the application is responsible for seeding the PRNG by calling *RAND_add()*, *RAND_egd(3)* or *RAND_load_file(3)*.

RAND_seed() is equivalent to *RAND_add()* when **num == entropy**.

RAND_event() and *RAND_screen()* are deprecated and should not be called.

RETURN VALUES

RAND_status() returns 1 if the PRNG has been seeded with enough data, 0 otherwise.

RAND_event() calls *RAND_poll()* and returns *RAND_status()*.

RAND_screen calls *RAND_poll()*.

The other functions do not return values.

HISTORY

RAND_event() and *RAND_screen()* are deprecated since OpenSSL 1.1.0. Use the functions described above instead.

SEE ALSO

rand(3), *RAND_egd(3)*, *RAND_load_file(3)*, *RAND_cleanup(3)*

COPYRIGHT