NAME

RAND_load_file, RAND_write_file, RAND_file_name - PRNG seed file

SYNOPSIS

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

int RAND_load_file(const char *filename, long max_bytes);

int RAND_write_file(const char *filename);

const char *RAND_file_name(char *buf, size_t num);
```

DESCRIPTION

RAND_load_file() reads a number of bytes from file **filename** and adds them to the PRNG. If **max_bytes** is nonnegative, up to **max_bytes** are read; if **max_bytes** is -1, the complete file is read. Do not load the same file multiple times unless its contents have been updated by **RAND_write_file()** between reads. Also, note that **filename** should be adequately protected so that an attacker cannot replace or examine the contents. If **filename** is not a regular file, then user is considered to be responsible for any side effects, e.g. non-anticipated blocking or capture of controlling terminal.

RAND_write_file() writes a number of random bytes (currently 128) to file **filename** which can be used to initialize the PRNG by calling **RAND_load_file()** in a later session.

RAND_file_name() generates a default path for the random seed file. **buf** points to a buffer of size **num** in which to store the filename.

On all systems, if the environment variable **RANDFILE** is set, its value will be used as the seed filename. Otherwise, the file is called ".rnd", found in platform dependent locations:

On Windows (in order of preference)

%HOME%, %USERPROFILE%, %SYSTEMROOT%, C:\

On VMS

SYS$LOGIN:

On all other systems

$HOME

If $HOME (on non-Windows and non-VMS system) is not set either, or **num** is too small for the pathname, an error occurs.

RETURN VALUES

RAND_load_file() returns the number of bytes read or -1 on error.

RAND_write_file() returns the number of bytes written, or -1 if the bytes written were generated without appropriate seeding.

RAND_file_name() returns a pointer to **buf** on success, and NULL on error.

SEE ALSO

RAND_add(3), **RAND_bytes**(3), **RAND**(7)

COPYRIGHT