NAME

SRP_VBASE_new, SRP_VBASE_free, SRP_VBASE_init, SRP_VBASE_add0_user, SRP_VBASE_get1_by_user, SRP_VBASE_get_by_user - Functions to create and manage a stack of SRP user verifier information

SYNOPSIS

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

The following functions have been deprecated since OpenSSL 3.0, and can be hidden entirely by defining **OPENSSL_API_COMPAT** with a suitable version value, see **openssl_user_macros**(7):

```
SRP_VBASE *SRP_VBASE_new(char *seed_key);
void SRP_VBASE_free(SRP_VBASE *vb);

int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file);

int SRP_VBASE_add0_user(SRP_VBASE *vb, SRP_user_pwd *user_pwd);
SRP_user_pwd *SRP_VBASE_get1_by_user(SRP_VBASE *vb, char *username);
SRP_user_pwd *SRP_VBASE_get_by_user(SRP_VBASE *vb, char *username);
```

DESCRIPTION

All of the functions described on this page are deprecated. There are no available replacement functions at this time.

The **SRP_VBASE_new()** function allocates a structure to store server side SRP verifier information. If **seed_key** is not NULL a copy is stored and used to generate dummy parameters for users that are not found by **SRP_VBASE_get1_by_user()**. This allows the server to hide the fact that it doesn't have a verifier for a particular username, as described in section 2.5.1.3 'Unknown SRP' of RFC 5054. The seed string should contain random NUL terminated binary data (therefore the random data should not contain NUL bytes!).

The **SRP_VBASE_free()** function frees up the **vb** structure. If **vb** is NULL, nothing is done.

The **SRP_VBASE_init()** function parses the information in a verifier file and populates the **vb** structure. The verifier file is a text file containing multiple entries, whose format is: flag base64(verifier) base64(salt) username gNid userinfo(optional) where the flag can be 'V' (valid) or 'R' (revoked). Note that the base64 encoding used here is non-standard so it is recommended to use **openssl-srp**(1) to generate this file.

The **SRP_VBASE_add0_user()** function adds the **user_pwd** verifier information to the **vb** structure. See

SRP_user_pwd_new(3) to create and populate this record. The library takes ownership of **user_pwd**, it should not be freed by the caller.

The **SRP_VBASE_get1_by_user()** function returns the password info for the user whose username matches **username**. It replaces the deprecated **SRP_VBASE_get_by_user()**. If no matching user is found but a seed_key and default gN parameters have been set, dummy authentication information is generated from the seed_key, allowing the server to hide the fact that it doesn't have a verifier for a particular username. When using SRP as a TLS authentication mechanism, this will cause the handshake to proceed normally but the first client will be rejected with a "bad_record_mac" alert, as if the password was incorrect. If no matching user is found and the seed_key is not set, NULL is returned. Ownership of the returned pointer is released to the caller, it must be freed with **SRP_user_pwd_free()**.

RETURN VALUES

SRP_VBASE_init() returns **SRP_NO_ERROR** (0) on success and a positive value on failure. The error codes are **SRP_ERR_OPEN_FILE** if the file could not be opened, **SRP_ERR_VBASE_INCOMPLETE_FILE** if the file could not be parsed, **SRP_ERR_MEMORY** on memory allocation failure and **SRP_ERR_VBASE_BN_LIB** for invalid decoded parameter values.

SRP_VBASE_add0_user() returns 1 on success and 0 on failure.

SEE ALSO

openssl-srp(1), **SRP_create_verifier**(3), **SRP_user_pwd_new**(3), **SSL_CTX_set_srp_password**(3)

HISTORY

The **SRP_VBASE_add0_user()** function was added in OpenSSL 3.0.

All other functions were added in OpenSSL 1.0.1.

All of these functions were deprecated in OpenSSL 3.0.

COPYRIGHT