NAME

Heimdal NTLM library -

Functions

void **heim_ntlm_free_buf** (struct **ntlm_buf** *p)
void **heim_ntlm_free_targetinfo** (struct ntlm_targetinfo *ti)
int **heim_ntlm_encode_targetinfo** (const struct ntlm_targetinfo *ti, int ucs2, struct **ntlm_buf** *data)
int **heim_ntlm_decode_targetinfo** (const struct **ntlm_buf** *data, int ucs2, struct ntlm_targetinfo *ti)
void **heim_ntlm_free_type1** (struct **ntlm_type1** *data)
int **heim_ntlm_encode_type1** (const struct **ntlm_type1** *type1, struct **ntlm_buf** *data)
void **heim_ntlm_free_type2** (struct **ntlm_type2** *data)
int **heim_ntlm_encode_type2** (const struct **ntlm_type2** *type2, struct **ntlm_buf** *data)
void **heim_ntlm_free_type3** (struct **ntlm_type3** *data)
int **heim_ntlm_encode_type3** (const struct **ntlm_type3** *type3, struct **ntlm_buf** *data)
int **heim_ntlm_nt_key** (const char *password, struct **ntlm_buf** *key)
int **heim_ntlm_calculate_ntlm1** (void *key, size_t len, unsigned char challenge[8], struct **ntlm_buf** *answer)
int **heim_ntlm_build_ntlm1_master** (void *key, size_t len, struct **ntlm_buf** *session, struct **ntlm_buf** *master)
int **heim_ntlm_build_ntlm2_master** (void *key, size_t len, struct **ntlm_buf** *blob, struct **ntlm_buf** *session, struct **ntlm_buf** *master)
int **heim_ntlm_keyex_unwrap** (struct **ntlm_buf** *baseKey, struct **ntlm_buf** *encryptedSession, struct **ntlm_buf** *session)
int **heim_ntlm_ntlmv2_key** (const void *key, size_t len, const char *username, const char *target, unsigned char ntlmv2[16])
int **heim_ntlm_calculate_lm2** (const void *key, size_t len, const char *username, const char *target, const unsigned char serverchallenge[8], unsigned char ntlmv2[16], struct **ntlm_buf** *answer)
int **heim_ntlm_calculate_ntlm2** (const void *key, size_t len, const char *username, const char *target, const unsigned char serverchallenge[8], const struct **ntlm_buf** *infotarget, unsigned char ntlmv2[16], struct **ntlm_buf** *answer)
int **heim_ntlm_verify_ntlm2** (const void *key, size_t len, const char *username, const char *target, time_t now, const unsigned char serverchallenge[8], const struct **ntlm_buf** *answer, struct **ntlm_buf** *infotarget, unsigned char ntlmv2[16])

Detailed Description

The NTLM core functions implement the string2key generation function, message encode and decode function, and the hash function functions.

Function Documentation

int heim_ntlm_build_ntlm1_master (void * key, size_t len, struct ntlm_buf * session, struct ntlm_buf * master)

Generates an NTLMv1 session random with assosited session master key.

Parameters:

key the ntlm v1 key
len length of key
session generated session nonce, should be freed with **heim_ntlm_free_buf()**.
master calculated session master key, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_build_ntlm2_master (void * key, size_t len, struct ntlm_buf * blob, struct ntlm_buf * session, struct ntlm_buf * master)

Generates an NTLMv2 session random with associated session master key.

Parameters:

key the NTLMv2 key
len length of key
blob the NTLMv2 'blob'
session generated session nonce, should be freed with **heim_ntlm_free_buf()**.
master calculated session master key, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_calculate_lm2 (const void * key, size_t len, const char * username, const char * target, const unsigned char serverchallenge[8], unsigned char ntlmv2[16], struct ntlm_buf * answer)

Calculate LMv2 response

Parameters:

key the ntlm key
len length of key
username name of the user, as sent in the message, assumed to be in UTF8.
target the name of the target, assumed to be in UTF8.
serverchallenge challenge as sent by the server in the type2 message.
ntlmv2 calculated session key
answer ntlm response answer, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_calculate_ntlm1 (void * key, size_t len, unsigned char challenge[8], struct ntlm_buf * answer)

Calculate NTLMv1 response hash

Parameters:

key the ntlm v1 key
len length of key
challenge sent by the server
answer calculated answer, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_calculate_ntlm2 (const void * key, size_t len, const char * username, const char * target, const unsigned char serverchallenge[8], const struct ntlm_buf * infotarget, unsigned char ntlmv2[16], struct ntlm_buf * answer)

Calculate NTLMv2 response

Parameters:

key the ntlm key
len length of key
username name of the user, as sent in the message, assumed to be in UTF8.
target the name of the target, assumed to be in UTF8.
serverchallenge challenge as sent by the server in the type2 message.
infotarget infotarget as sent by the server in the type2 message.
ntlmv2 calculated session key
answer ntlm response answer, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_decode_targetinfo (const struct ntlm_buf * data, int ucs2, struct ntlm_targetinfo * ti)

Decodes an NTLM targetinfo message

Parameters:

data input data buffer with the encode NTLM targetinfo message
ucs2 if the strings should be encoded with ucs2 (selected by flag in message).
ti the decoded target info, should be freed with **heim_ntlm_free_targetinfo()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_encode_targetinfo (const struct ntlm_targetinfo * ti, int ucs2, struct ntlm_buf * data)

Encodes a ntlm_targetinfo message.

Parameters:

ti the ntlm_targetinfo message to encode.
ucs2 ignored
data is the return buffer with the encoded message, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_encode_type1 (const struct ntlm_type1 * type1, struct ntlm_buf * data)

Encodes an **ntlm_type1** message.

Parameters:

type1 the **ntlm_type1** message to encode.
data is the return buffer with the encoded message, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_encode_type2 (const struct ntlm_type2 * type2, struct ntlm_buf * data)

Encodes an **ntlm_type2** message.

Parameters:

type2 the **ntlm_type2** message to encode.
data is the return buffer with the encoded message, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_encode_type3 (const struct ntlm_type3 * type3, struct ntlm_buf * data)

Encodes an **ntlm_type3** message.

Parameters:

type3 the **ntlm_type3** message to encode.
data is the return buffer with the encoded message, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

void heim_ntlm_free_buf (struct ntlm_buf * p)

heim_ntlm_free_buf frees the ntlm buffer

Parameters:

p buffer to be freed

void heim_ntlm_free_targetinfo (struct ntlm_targetinfo * ti)

Frees the ntlm_targetinfo message

Parameters:

ti targetinfo to be freed

void heim_ntlm_free_type1 (struct ntlm_type1 * data)

Frees the **ntlm_type1** message

Parameters:

data message to be freed

void heim_ntlm_free_type2 (struct ntlm_type2 * data)

Frees the **ntlm_type2** message

Parameters:

data message to be freed

void heim_ntlm_free_type3 (struct ntlm_type3 * data)

Frees the **ntlm_type3** message

Parameters:

data message to be freed

int heim_ntlm_keyex_unwrap (struct ntlm_buf * baseKey, struct ntlm_buf * encryptedSession, struct ntlm_buf * session)

Given a key and encrypted session, unwrap the session key

Parameters:

baseKey the sessionBaseKey

encryptedSession encrypted session, type3.session field.

session generated session nonce, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_nt_key (const char * password, struct ntlm_buf * key)

Calculate the NTLM key, the password is assumed to be in UTF8.

Parameters:

password password to calcute the key for.

key calcuted key, should be freed with **heim_ntlm_free_buf()**.

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.

int heim_ntlm_ntlmv2_key (const void * key, size_t len, const char * username, const char * target, unsigned char ntlmv2[16])

Generates an NTLMv2 session key.

Parameters:

key the ntlm key

len length of key

username name of the user, as sent in the message, assumed to be in UTF8.

target the name of the target, assumed to be in UTF8.

ntlmv2 the ntlmv2 session key

Returns:

0 on success, or an error code on failure.

int heim_ntlm_verify_ntlm2 (const void * key, size_t len, const char * username, const char * target, time_t now, const unsigned char serverchallenge[8], const struct ntlm_buf * answer, struct ntlm_buf * infotarget, unsigned char ntlmv2[16])

Verify NTLMv2 response.

Parameters:

key the ntlm key

len length of key

username name of the user, as sent in the message, assumed to be in UTF8.

target the name of the target, assumed to be in UTF8.

now the time now (0 if the library should pick it up itself)

serverchallenge challenge as sent by the server in the type2 message.
answer ntlm response answer, should be freed with **heim_ntlm_free_buf()**.
infotarget infotarget as sent by the server in the type2 message.
ntlmv2 calculated session key

Returns:

In case of success 0 is return, an errors, a errno in what went wrong.