NAME

ldns_dane_create_tlsa_rr, ldns_dane_create_tlsa_owner, ldns_dane_cert2rdf, ldns_dane_select_certificate - TLSA RR creation functions

SYNOPSIS

```
#include <stdint.h>
#include <stdbool.h>

#include <ldns/ldns.h>

ldns_status ldns_dane_create_tlsa_rr(ldns_rr** tlsa, ldns_tlsa_certificate_usage certificate_usage,
ldns_tlsa_selector selector, ldns_tlsa_matching_type matching_type, X509* cert);

ldns_status ldns_dane_create_tlsa_owner(ldns_rdf** tlsa_owner, const ldns_rdf* name, uint16_t port,
ldns_dane_transport transport);

ldns_status ldns_dane_cert2rdf(ldns_rdf** rdf, X509* cert, ldns_tlsa_selector selector,
ldns_tlsa_matching_type matching_type);

ldns_status ldns_dane_select_certificate(X509** selected_cert, X509* cert, STACK_OF(X509)*
extra_certs, X509_STORE* pkix_validation_store, ldns_tlsa_certificate_usage cert_usage, int index);
```

DESCRIPTION

ldns_dane_create_tlsa_rr() Creates a TLSA resource record from the certificate. No PKIX validation is performed! The given certificate is used as data regardless the value of certificate_usage.

tlsa: The created TLSA resource record.
certificate_usage: The value for the Certificate Usage field
selector: The value for the Selector field
matching_type: The value for the Matching Type field
cert: The certificate which data will be represented

Returns LDNS_STATUS_OK on success or an error code otherwise.

ldns_dane_create_tlsa_owner() Creates a dname consisting of the given name, prefixed by the service port and type of transport: _<EM>port</EM>._<EM>transport</EM>.<EM>name</EM>.

tlsa_owner: The created dname.
name: The dname that should be prefixed.

port: The service port number for which the name should be created.
transport: The transport for which the name should be created.
Returns LDNS_STATUS_OK on success or an error code otherwise.

ldns_dane_cert2rdf() Creates a LDNS_RDF_TYPE_HEX type rdf based on the binary data chosen by the selector and encoded using matching_type.

rdf: The created created rdf of type LDNS_RDF_TYPE_HEX.
cert: The certificate from which the data is selected
selector: The full certificate or the public key
matching_type: The full data or the SHA256 or SHA512 hash of the selected data
Returns LDNS_STATUS_OK on success or an error code otherwise.

ldns_dane_select_certificate() Selects the certificate from cert, extra_certs or the pkix_validation_store based on the value of cert_usage and index.

selected_cert: The selected cert.
cert: The certificate to validate (or not)
extra_certs: Intermediate certificates that might be necessary during validation. May be NULL, except when the certificate usage is "Trust Anchor Assertion" because the trust anchor has to be provided.(otherwise choose a "Domain issued certificate!"
pkix_validation_store: Used when the certificate usage is "CA constraint" or "Service Certificate Constraint" to validate the certificate and, in case of "CA constraint", select the CA. When pkix_validation_store is NULL, validation is explicitly turned off and the behaviour is then the same as for "Trust anchor assertion" and "Domain issued certificate" respectively.
cert_usage: Which certificate to use and how to validate.
index: Used to select the trust anchor when certificate usage is "Trust Anchor Assertion". 0 is the last certificate in the validation chain. 1 the one but last, etc. When index is -1, the last certificate is used that MUST be self-signed. This can help to make sure that the intended (self signed) trust anchor is actually present in extra_certs (which is a DANE requirement).

Returns LDNS_STATUS_OK on success or an error code otherwise.

AUTHOR

The ldns team at NLnet Labs.

REPORTING BUGS

Please report bugs to ldns-team@nlnetlabs.nl or in our bugzilla at
http://www.nlnetlabs.nl/bugs/index.html

COPYRIGHT

SEE ALSO

ldns_dane_verify, *ldns_dane_verify_rr*. And **perldoc Net::DNS**, **RFC1034**, **RFC1035**, **RFC4033**, **RFC4034** and **RFC4035**.

REMARKS

This manpage was automatically generated from the ldns source code.