## NAME

ldns_dane_verify, ldns_dane_verify_rr - TLSA RR verification functions

## SYNOPSIS

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

#include <ldns/ldns.h>

ldns_status ldns_dane_verify(const ldns_rr_list* tlsas, X509* cert, STACK_OF(X509)* extra_certs,
X509_STORE* pkix_validation_store);

ldns_status ldns_dane_verify_rr(const ldns_rr* tlsa_rr, X509* cert, STACK_OF(X509)* extra_certs,
X509_STORE* pkix_validation_store);
```

## DESCRIPTION

*ldns_dane_verify*() BEWARE! We strongly recommend to use OpenSSL 1.1.0 dane verification functions instead of the ones provided by ldns. When OpenSSL 1.1.0 was available ldns will use the OpenSSL 1.1.0 dane verification functions under the hood. When ldns was linked with OpenSSL < 1.1.0, this function will not be able to verify TLSA records with DANE-TA usage types.

BEWARE! The ldns dane verification functions do *not* do server name checks. The user has to perform additional server name checks themselves!

Verify if any of the given TLSA resource records matches the given certificate.

**tlsas**: The resource records that specify what and how to match the certificate. One must match for this function to succeed. With tlsas == NULL or the number of TLSA records in tlsas == 0, regular PKIX validation is performed.
**cert**: The certificate to match (and validate)
**extra_certs**: Intermediate certificates that might be necessary creating the validation chain.
**pkix_validation_store**: Used when the certificate usage is "CA constraint" or "Service Certificate Constraint" to validate the certificate.

Returns LDNS_STATUS_OK on success,
LDNS_STATUS_DANE_NEED_OPENSSL_GE_1_1_FOR_DANE_TA when at least one of the TLSA’s had usage type DANE-TA and none of the TLSA’s matched or PKIX validated,
LDNS_STATUS_DANE_PKIX_DID_NOT_VALIDATE when one of the TLSA’s matched but

the PKIX validation failed, LDNS_STATUS_DANE_TLSA_DID_NOT_MATCH when none of the TLSA's matched, or other ldns_status errors.

*ldns_dane_verify_rr*() BEWARE! We strongly recommend to use OpenSSL 1.1.0 dane verification functions instead of the ones provided by ldns. When OpenSSL 1.1.0 was available ldns will use the OpenSSL 1.1.0 dane verification functions under the hood. When ldns was linked with OpenSSL < 1.1.0, this function will not be able to verify TLSA records with DANE-TA usage types.

BEWARE! The ldns dane verification functions do *not* do server name checks. The user has to perform additional server name checks themselves!

Verify if the given TLSA resource record matches the given certificate. Reporting on a TLSA rr mismatch (LDNS_STATUS_DANE_TLSA_DID_NOT_MATCH) is preferred over PKIX failure (LDNS_STATUS_DANE_PKIX_DID_NOT_VALIDATE). So when PKIX validation is required by the TLSA Certificate usage, but the TLSA data does not match, LDNS_STATUS_DANE_TLSA_DID_NOT_MATCH is returned whether the PKIX validated or not.

When ldns is linked with OpenSSL < 1.1.0 and this function is available, then the DANE-TA usage type will not be verified, and on a tlsa_rr with this usage type, LDNS_STATUS_DANE_NEED_OPENSSL_GE_1_1_FOR_DANE_TA will be returned.

**tlsa_rr**: The resource record that specifies what and how to match the certificate. With tlsa_rr == NULL, regular PKIX validation is performed.
**cert**: The certificate to match (and validate)
**extra_certs**: Intermediate certificates that might be necessary creating the validation chain.
**pkix_validation_store**: Used when the certificate usage is "CA constraint" or "Service Certificate Constraint" to validate the certificate.

Returns LDNS_STATUS_OK on success, LDNS_STATUS_DANE_NEED_OPENSSL_GE_1_1_FOR_DANE_TA when the provided TLSA had the DANE-TA usage type, LDNS_STATUS_DANE_TLSA_DID_NOT_MATCH on TLSA data mismatch, LDNS_STATUS_DANE_PKIX_DID_NOT_VALIDATE when TLSA matched, but the PKIX validation failed, or other ldns_status errors.

## 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_create_tlsa_owner*, *ldns_dane_cert2rdf*, *ldns_dane_select_certificate*, *ldns_dane_create_tlsa_rr*. And **perldoc Net::DNS**, **RFC1034**, **RFC1035**, **RFC4033**, **RFC4034** and **RFC4035**.

## REMARKS

This manpage was automatically generated from the ldns source code.