Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'SCT_get_validation_status.3ossl' command

$ man SCT_get_validation_status.3ossl

SCT_VALIDATE(3ossl) OpenSSL SCT_VALIDATE(3ossl)

NAME

SCT_validate, SCT_LIST_validate, SCT_get_validation_status - checks Signed Certificate Timestamps (SCTs) are valid

SYNOPSIS

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

typedef enum {
    SCT_VALIDATION_STATUS_NOT_SET,
    SCT_VALIDATION_STATUS_UNKNOWN_LOG,
    SCT_VALIDATION_STATUS_VALID,
    SCT_VALIDATION_STATUS_INVALID,
    SCT_VALIDATION_STATUS_UNVERIFIED,
    SCT_VALIDATION_STATUS_UNKNOWN_VERSION
} sct_validation_status_t;

int SCT_validate(SCT *sct, const CT_POLICY_EVAL_CTX *ctx);
int SCT_LIST_validate(const STACK_OF(SCT) *scts, CT_POLICY_EVAL_CTX *ctx);
sct_validation_status_t SCT_get_validation_status(const SCT *sct);
```

DESCRIPTION

SCT_validate() will check that an SCT is valid and verify its signature. SCT_LIST_validate() performs the same checks on an entire stack of SCTs. The result of the validation checks can be obtained by passing the SCT to SCT_get_validation_status().

A CT_POLICY_EVAL_CTX must be provided that specifies:

? The certificate the SCT was issued for.

Failure to provide the certificate will result in the validation status being SCT_VALIDATION_STATUS_UNVERIFIED.

? The issuer of that certificate.

This is only required if the SCT was issued for a pre-certificate (see RFC 6962). If it is required but not provided, the validation status will be SCT_VALIDATION_STATUS_UNVERIFIED.

? A CTLOG_STORE that contains the CT log that issued this SCT.

If the SCT was issued by a log that is not in this CTLOG_STORE, the validation status will be SCT_VALIDATION_STATUS_UNKNOWN_LOG.

If the SCT is of an unsupported version (only v1 is currently supported), the validation status will be SCT_VALIDATION_STATUS_UNKNOWN_VERSION.

If the SCT's signature is incorrect, its timestamp is in the future (relative to the time in CT_POLICY_EVAL_CTX), or if it is otherwise invalid, the validation status will be SCT_VALIDATION_STATUS_INVALID.

If all checks pass, the validation status will be SCT_VALIDATION_STATUS_VALID.

NOTES

A return value of 0 from SCT_LIST_validate() should not be interpreted as a failure. At a minimum, only one valid SCT may provide sufficient confidence that a certificate has been publicly logged.

RETURN VALUES

SCT_validate() returns a negative integer if an internal error occurs, 0 if the SCT fails validation, or 1 if the SCT passes validation.

SCT_LIST_validate() returns a negative integer if an internal error occurs, 0 if any of SCTs fails validation, or 1 if they all pass validation.

SCT_get_validation_status() returns the validation status of the SCT. If SCT_validate() or SCT_LIST_validate() have not been passed that SCT, the returned value will be SCT_VALIDATION_STATUS_NOT_SET.

SEE ALSO

ct(7)

HISTORY

These functions were added in OpenSSL 1.1.0.

COPYRIGHT