NAME

crlutil − List, generate, modify, or delete CRLs within the NSS security database file(s) and list, create, modify or delete certificates entries in a particular CRL.

SYNOPSIS

crlutil [*options*] [[*arguments*]]

STATUS

This documentation is still work in progress. Please contribute to the initial review in **Mozilla NSS bug 836477**[1]

DESCRIPTION

The Certificate Revocation List (CRL) Management Tool, **crlutil**, is a command−line utility that can list, generate, modify, or delete CRLs within the NSS security database file(s) and list, create, modify or delete certificates entries in a particular CRL.

The key and certificate management process generally begins with creating keys in the key database, then generating and managing certificates in the certificate database(see certutil tool) and continues with certificates expiration or revocation.

This document discusses certificate revocation list management. For information on security module database management, see Using the Security Module Database Tool. For information on certificate and key database management, see Using the Certificate Database Tool.

To run the Certificate Revocation List Management Tool, type the command

crlutil option [arguments]

where options and arguments are combinations of the options and arguments listed in the following section. Each command takes one option. Each option may take zero or more arguments. To see a usage string, issue the command without options, or with the −H option.

OPTIONS AND ARGUMENTS

Options

Options specify an action. Option arguments modify an action. The options and arguments for the crlutil command are defined as follows:

−D
: Delete Certificate Revocation List from cert database.

−E
: Erase all CRLs of specified type from the cert database

−G
: Create new Certificate Revocation List (CRL).

−I
: Import a CRL to the cert database

−L
: List existing CRL located in cert database file.

−M
: Modify existing CRL which can be located in cert db or in arbitrary file. If located in file it should be encoded in ASN.1 encode format.

−S
: Show contents of a CRL file which isn't stored in the database.

Arguments

Option arguments modify an action.

−a

Use ASCII format or allow the use of ASCII format for input and output. This formatting follows RFC #1113.

−B
Bypass CA signature checks.

−c crl−gen−file
Specify script file that will be used to control crl generation/modification. See crl−cript−file format below. If options −M|−G is used and −c crl−script−file is not specified, crlutil will read script data from standard input.

−d directory
Specify the database directory containing the certificate and key database files. On Unix the Certificate Database Tool defaults to $HOME/.netscape (that is, ˜/.netscape). On Windows NT the default is the current directory.

The NSS database files must reside in the same directory.

−f password−file
Specify a file that will automatically supply the password to include in a certificate or to access a certificate database. This is a plain−text file containing one password. Be sure to prevent unauthorized access to this file.

−i crl−file
Specify the file which contains the CRL to import or show.

−l algorithm−name
Specify a specific signature algorithm. List of possible algorithms: MD2 | MD4 | MD5 | SHA1 | SHA256 | SHA384 | SHA512

−n nickname
Specify the nickname of a certificate or key to list, create, add to a database, modify, or validate. Bracket the nickname string with quotation marks if it contains spaces.

−o output−file
Specify the output file name for new CRL. Bracket the output−file string with quotation marks if it contains spaces. If this argument is not used the output destination defaults to standard output.

−P dbprefix
Specify the prefix used on the NSS security database files (for example, my_cert8.db and my_key3.db). This option is provided as a special case. Changing the names of the certificate and key databases is not recommended.

−t crl−type
Specify type of CRL. possible types are: 0 − SEC_KRL_TYPE, 1 − SEC_CRL_TYPE. This option is obsolete

−u url
Specify the url.

−w pwd−string
Provide db password in command line.

−Z algorithm
Specify the hash algorithm to use for signing the CRL.

CRL GENERATION SCRIPT SYNTAX

CRL generation script file has the following syntax:

* Line with comments should have # as a first symbol of a line

* Set "this update" or "next update" CRL fields:

update=YYYYMMDDhhmmssZ nextupdate=YYYYMMDDhhmmssZ

Field "next update" is optional. Time should be in GeneralizedTime format (YYYYMMDDhhmmssZ). For example: 20050204153000Z

* Add an extension to a CRL or a crl certificate entry:

addext extension−name critical/non−critical [arg1[arg2 ...]]

Where:

extension−name: string value of a name of known extensions. critical/non−critical: is 1 when extension is critical and 0 otherwise. arg1, arg2: specific to extension type extension parameters

addext uses the range that was set earlier by addcert and will install an extension to every cert entries within the range.

* Add certificate entries(s) to CRL:

addcert range date

range: two integer values separated by dash: range of certificates that will be added by this command. dash is used as a delimiter. Only one cert will be added if there is no delimiter. date: revocation date of a cert. Date should be represented in GeneralizedTime format (YYYYMMDDhhmmssZ).

* Remove certificate entry(s) from CRL

rmcert range

Where:

range: two integer values separated by dash: range of certificates that will be added by this command. dash is used as a delimiter. Only one cert will be added if there is no delimiter.

* Change range of certificate entry(s) in CRL

range new−range

Where:

new−range: two integer values separated by dash: range of certificates that will be added by this command. dash is used as a delimiter. Only one cert will be added if there is no delimiter.

Implemented Extensions

The extensions defined for CRL provide methods for associating additional attributes with CRLs of theirs entries. For more information see RFC #3280

* Add The Authority Key Identifier extension:

The authority key identifier extension provides a means of identifying the public key corresponding to the private key used to sign a CRL.

authKeyId critical [key−id | dn cert−serial]

Where:

authKeyIdent: identifies the name of an extension critical: value of 1 of 0. Should be set to 1 if this extension is critical or 0 otherwise. key−id: key identifier represented in octet string. dn:: is a CA distinguished name cert−serial: authority certificate serial number.

* Add Issuer Alternative Name extension:

The issuer alternative names extension allows additional identities to be associated with the issuer of the CRL. Defined options include an rfc822 name (electronic mail address), a DNS name, an IP address, and a URI.

issuerAltNames non−critical name−list

Where:

subjAltNames: identifies the name of an extension should be set to 0 since this is non−critical extension name−list: comma separated list of names

* Add CRL Number extension:

The CRL number is a non−critical CRL extension which conveys a monotonically increasing sequence number for a given CRL scope and CRL issuer. This extension allows users to easily determine when a particular CRL supersedes another CRL

crlNumber non−critical number

Where:

crlNumber: identifies the name of an extension critical: should be set to 0 since this is non−critical extension number: value of long which identifies the sequential number of a CRL.

* Add Revocation Reason Code extension:

The reasonCode is a non−critical CRL entry extension that identifies the reason for the certificate revocation.

reasonCode non−critical code

Where:

reasonCode: identifies the name of an extension non−critical: should be set to 0 since this is non−critical extension code: the following codes are available:

unspecified (0), keyCompromise (1), cACompromise (2), affiliationChanged (3), superseded (4), cessationOfOperation (5), certificateHold (6), removeFromCRL (8), privilegeWithdrawn (9), aACompromise (10)

* Add Invalidity Date extension:

The invalidity date is a non−critical CRL entry extension that provides the date on which it is known or suspected that the private key was compromised or that the certificate otherwise became invalid.

invalidityDate non−critical date

Where:

crlNumber: identifies the name of an extension non−critical: should be set to 0 since this is non−critical extension date: invalidity date of a cert. Date should be represented in GeneralizedTime format (YYYYMMDDhhmmssZ).

USAGE

The Certificate Revocation List Management Tool's capabilities are grouped as follows, using these combinations of options and arguments. Options and arguments in square brackets are optional, those without square brackets are required.

See "Implemented extensions" for more information regarding extensions and their parameters.

* Creating or modifying a CRL:

crlutil −G|−M −c crl−gen−file −n nickname [−i crl] [−u url] [−d keydir] [−P dbprefix] [−l alg] [−a] [−B]

* Listing all CRls or a named CRL:

crlutil −L [−n crl−name] [−d krydir]

* Deleting CRL from db:

crlutil −D −n nickname [−d keydir] [−P dbprefix]

* Erasing CRLs from db:

crlutil −E [−d keydir] [−P dbprefix]

* Deleting CRL from db:

crlutil −D −n nickname [−d keydir] [−P dbprefix]

* Erasing CRLs from db:

crlutil −E [−d keydir] [−P dbprefix]

* Import CRL from file:

crlutil −I −i crl [−t crlType] [−u url] [−d keydir] [−P dbprefix] [−B]

SEE ALSO

certutil(1)

ADDITIONAL RESOURCES

For information about NSS and other tools related to NSS (like JSS), check out the NSS project wiki at **http://www.mozilla.org/projects/security/pki/nss/**. The NSS site relates directly to NSS code changes and releases.

Mailing lists: https://lists.mozilla.org/listinfo/dev−tech−crypto

IRC: Freenode at #dogtag−pki

AUTHORS

The NSS tools were written and maintained by developers with Netscape, Red Hat, Sun, Oracle, Mozilla, and Google.

Authors: Elio Maldonado <emaldona@redhat.com>, Deon Lackey <dlackey@redhat.com>.

LICENSE

NOTES

1. Mozilla NSS bug 836477
   https://bugzilla.mozilla.org/show_bug.cgi?id=836477