NAME

openssl-ocsp - Online Certificate Status Protocol command

SYNOPSIS

OCSP Client

openssl ocsp [**-help**] [**-out** *file*] [**-issuer** *file*] [**-cert** *file*] [**-no_certs**] [**-serial** *n*] [**-signer** *file*] [**-signkey** *file*] [**-sign_other** *file*] [**-nonce**] [**-no_nonce**] [**-req_text**] [**-resp_text**] [**-text**] [**-reqout** *file*] [**-respout** *file*] [**-reqin** *file*] [**-respin** *file*] [**-url** *URL*] [**-host** *host*:*port*] [**-path**] [**-proxy** *[http[s]://][userinfo@]host[:port][/path]*] [**-no_proxy** *addresses*] [**-header**] [**-timeout** *seconds*] [**-VAfile** *file*] [**-validity_period** *n*] [**-status_age** *n*] [**-noverify**] [**-verify_other** *file*] [**-trust_other**] [**-no_intern**] [**-no_signature_verify**] [**-no_cert_verify**] [**-no_chain**] [**-no_cert_checks**] [**-no_explicit**] [**-port** *num*] [**-ignore_err**]

OCSP Server

openssl ocsp [**-index** *file*] [**-CA** *file*] [**-rsigner** *file*] [**-rkey** *file*] [**-passin** *arg*] [**-rother** *file*] [**-rsigopt** *nm*:*v*] [**-rmd** *digest*] [**-badsig**] [**-resp_no_certs**] [**-nmin** *n*] [**-ndays** *n*] [**-resp_key_id**] [**-nrequest** *n*] [**-multi** *process-count*] [**-rcid** *digest*] [***-digest***] [**-CAfile** *file*] [**-no-CAfile**] [**-CApath** *dir*] [**-no-CApath**] [**-CAstore** *uri*] [**-no-CAstore**] [**-allow_proxy_certs**] [**-attime** *timestamp*] [**-no_check_time**] [**-check_ss_sig**] [**-crl_check**] [**-crl_check_all**] [**-explicit_policy**] [**-extended_crl**] [**-ignore_critical**] [**-inhibit_any**] [**-inhibit_map**] [**-partial_chain**] [**-policy** *arg*] [**-policy_check**] [**-policy_print**] [**-purpose** *purpose*] [**-suiteB_128**] [**-suiteB_128_only**] [**-suiteB_192**] [**-trusted_first**] [**-no_alt_chains**] [**-use_deltas**] [**-auth_level** *num*] [**-verify_depth** *num*] [**-verify_email** *email*] [**-verify_hostname** *hostname*] [**-verify_ip** *ip*] [**-verify_name** *name*] [**-x509_strict**] [**-issuer_checks**] [**-provider** *name*] [**-provider-path** *path*] [**-propquery** *propq*]

DESCRIPTION

The Online Certificate Status Protocol (OCSP) enables applications to determine the (revocation) state of an identified certificate (RFC 2560).

This command performs many common OCSP tasks. It can be used to print out requests and responses, create requests and send queries to an OCSP responder and behave like a mini OCSP server itself.

OPTIONS

This command operates as either a client or a server. The options are described below, divided into those two modes.

OCSP Client Options

-help

Print out a usage message.

-out *filename*
specify output filename, default is standard output.

-issuer *filename*
This specifies the current issuer certificate. This option can be used multiple times. This option **MUST** come before any **-cert** options.

-cert *filename*
Add the certificate *filename* to the request. The issuer certificate is taken from the previous **-issuer** option, or an error occurs if no issuer certificate is specified.

-no_certs
Don’t include any certificates in signed request.

-serial *num*
Same as the **-cert** option except the certificate with serial number **num** is added to the request. The serial number is interpreted as a decimal integer unless preceded by "0x". Negative integers can also be specified by preceding the value by a "-" sign.

-signer *filename*, **-signkey** *filename*
Sign the OCSP request using the certificate specified in the **-signer** option and the private key specified by the **-signkey** option. If the **-signkey** option is not present then the private key is read from the same file as the certificate. If neither option is specified then the OCSP request is not signed.

-sign_other *filename*
Additional certificates to include in the signed request. The input can be in PEM, DER, or PKCS#12 format.

-nonce, **-no_nonce**
Add an OCSP nonce extension to a request or disable OCSP nonce addition. Normally if an OCSP request is input using the **-reqin** option no nonce is added: using the **-nonce** option will force addition of a nonce. If an OCSP request is being created (using **-cert** and **-serial** options) a nonce is automatically added specifying **-no_nonce** overrides this.

-req_text, **-resp_text**, **-text**
Print out the text form of the OCSP request, response or both respectively.

-reqout *file*, **-respout** *file*
Write out the DER encoded certificate request or response to *file*.

-reqin *file*, **-respin** *file*

Read OCSP request or response file from *file*. These option are ignored if OCSP request or response creation is implied by other options (for example with **-serial**, **-cert** and **-host** options).

-url *responder_url*

Specify the responder URL. Both HTTP and HTTPS (SSL/TLS) URLs can be specified. The optional userinfo and fragment components are ignored. Any given query component is handled as part of the path component.

-host *hostname*:*port*, **-path** *pathname*

If the **-host** option is present then the OCSP request is sent to the host *hostname* on port *port*. The **-path** option specifies the HTTP pathname to use or "/" by default. This is equivalent to specifying **-url** with scheme http:// and the given hostname, port, and pathname.

-proxy *[http[s]://][userinfo@]host[:port][/path]*

The HTTP(S) proxy server to use for reaching the OCSP server unless **-no_proxy** applies, see below. The proxy port defaults to 80 or 443 if the scheme is "https"; apart from that the optional "http://" or "https://" prefix is ignored, as well as any userinfo and path components. Defaults to the environment variable "http_proxy" if set, else "HTTP_PROXY" in case no TLS is used, otherwise "https_proxy" if set, else "HTTPS_PROXY".

-no_proxy *addresses*

List of IP addresses and/or DNS names of servers not to use an HTTP(S) proxy for, separated by commas and/or whitespace (where in the latter case the whole argument must be enclosed in "..."). Default is from the environment variable "no_proxy" if set, else "NO_PROXY".

-header *name=value*

Adds the header *name* with the specified *value* to the OCSP request that is sent to the responder. This may be repeated.

-timeout *seconds*

Connection timeout to the OCSP responder in seconds. On POSIX systems, when running as an OCSP responder, this option also limits the time that the responder is willing to wait for the client request. This time is measured from the time the responder accepts the connection until the complete request is received.

-verify_other *file*

File or URI containing additional certificates to search when attempting to locate the OCSP response signing certificate. Some responders omit the actual signer’s certificate from the response: this option can be used to supply the necessary certificate in such cases. The input can

be in PEM, DER, or PKCS#12 format.

-trust_other
The certificates specified by the **-verify_other** option should be explicitly trusted and no additional checks will be performed on them. This is useful when the complete responder certificate chain is not available or trusting a root CA is not appropriate.

-VAfile *file*
File or URI containing explicitly trusted responder certificates. Equivalent to the **-verify_other** and **-trust_other** options. The input can be in PEM, DER, or PKCS#12 format.

-noverify
Don't attempt to verify the OCSP response signature or the nonce values. This option will normally only be used for debugging since it disables all verification of the responders certificate.

-no_intern
Ignore certificates contained in the OCSP response when searching for the signers certificate. With this option the signers certificate must be specified with either the **-verify_other** or **-VAfile** options.

-no_signature_verify
Don't check the signature on the OCSP response. Since this option tolerates invalid signatures on OCSP responses it will normally only be used for testing purposes.

-no_cert_verify
Don't verify the OCSP response signers certificate at all. Since this option allows the OCSP response to be signed by any certificate it should only be used for testing purposes.

-no_chain
Do not use certificates in the response as additional untrusted CA certificates.

-no_explicit
Do not explicitly trust the root CA if it is set to be trusted for OCSP signing.

-no_cert_checks
Don't perform any additional checks on the OCSP response signers certificate. That is do not make any checks to see if the signers certificate is authorised to provide the necessary status information: as a result this option should only be used for testing purposes.

-validity_period *nsec*, **-status_age** *age*

These options specify the range of times, in seconds, which will be tolerated in an OCSP response. Each certificate status response includes a **notBefore** time and an optional **notAfter** time. The current time should fall between these two values, but the interval between the two times may be only a few seconds. In practice the OCSP responder and clients clocks may not be precisely synchronised and so such a check may fail. To avoid this the **-validity_period** option can be used to specify an acceptable error range in seconds, the default value is 5 minutes.

If the **notAfter** time is omitted from a response then this means that new status information is immediately available. In this case the age of the **notBefore** field is checked to see it is not older than *age* seconds old. By default this additional check is not performed.

-rcid *digest*

This option sets the digest algorithm to use for certificate identification in the OCSP response. Any digest supported by the **openssl-dgst**(1) command can be used. The default is the same digest algorithm used in the request.

-digest

This option sets digest algorithm to use for certificate identification in the OCSP request. Any digest supported by the OpenSSL **dgst** command can be used. The default is SHA-1. This option may be used multiple times to specify the digest used by subsequent certificate identifiers.

-CAfile *file*, **-no-CAfile**, **-CApath** *dir*, **-no-CApath**, **-CAstore** *uri*, **-no-CAstore**

See "Trusted Certificate Options" in **openssl-verification-options**(1) for details.

-allow_proxy_certs, **-attime**, **-no_check_time**, **-check_ss_sig**, **-crl_check**, **-crl_check_all**, **-explicit_policy**, **-extended_crl**, **-ignore_critical**, **-inhibit_any**, **-inhibit_map**, **-no_alt_chains**, **-partial_chain**, **-policy**, **-policy_check**, **-policy_print**, **-purpose**, **-suiteB_128**, **-suiteB_128_only**, **-suiteB_192**, **-trusted_first**, **-use_deltas**, **-auth_level**, **-verify_depth**, **-verify_email**, **-verify_hostname**, **-verify_ip**, **-verify_name**, **-x509_strict** **-issuer_checks**

Set various options of certificate chain verification. See "Verification Options" in **openssl-verification-options**(1) for details.

-provider *name*
-provider-path *path*
-propquery *propq*

See "Provider Options" in **openssl**(1), **provider**(7), and **property**(7).

OCSP Server Options

-index *indexfile*

The *indexfile* parameter is the name of a text index file in **ca** format containing certificate

revocation information.

If the **-index** option is specified then this command switches to responder mode, otherwise it is in client mode. The request(s) the responder processes can be either specified on the command line (using **-issuer** and **-serial** options), supplied in a file (using the **-reqin** option) or via external OCSP clients (if **-port** or **-url** is specified).

If the **-index** option is present then the **-CA** and **-rsigner** options must also be present.

-CA *file*
CA certificate corresponding to the revocation information in the index file given with **-index**. The input can be in PEM, DER, or PKCS#12 format.

-rsigner *file*
The certificate to sign OCSP responses with.

-rkey *file*
The private key to sign OCSP responses with: if not present the file specified in the **-rsigner** option is used.

-passin *arg*
The private key password source. For more information about the format of *arg* see **openssl-passphrase-options**(1).

-rother *file*
Additional certificates to include in the OCSP response. The input can be in PEM, DER, or PKCS#12 format.

-rsigopt *nm*:*v*
Pass options to the signature algorithm when signing OCSP responses. Names and values of these options are algorithm-specific.

-rmd *digest*
The digest to use when signing the response.

-badsig
Corrupt the response signature before writing it; this can be useful for testing.

-resp_no_certs
Don’t include any certificates in the OCSP response.

-resp_key_id
Identify the signer certificate using the key ID, default is to use the subject name.

-port *portnum*
Port to listen for OCSP requests on. The port may also be specified using the **url** option. A 0 argument indicates that any available port shall be chosen automatically.

-ignore_err
Ignore malformed requests or responses: When acting as an OCSP client, retry if a malformed response is received. When acting as an OCSP responder, continue running instead of terminating upon receiving a malformed request.

-nrequest *number*
The OCSP server will exit after receiving *number* requests, default unlimited.

-multi *process-count*
Run the specified number of OCSP responder child processes, with the parent process respawning child processes as needed. Child processes will detect changes in the CA index file and automatically reload it. When running as a responder **-timeout** option is recommended to limit the time each child is willing to wait for the client's OCSP response. This option is available on POSIX systems (that support the **fork()** and other required unix system-calls).

-nmin *minutes*, **-ndays** *days*
Number of minutes or days when fresh revocation information is available: used in the **nextUpdate** field. If neither option is present then the **nextUpdate** field is omitted meaning fresh revocation information is immediately available.

OCSP RESPONSE VERIFICATION

OCSP Response follows the rules specified in RFC2560.

Initially the OCSP responder certificate is located and the signature on the OCSP request checked using the responder certificate's public key.

Then a normal certificate verify is performed on the OCSP responder certificate building up a certificate chain in the process. The locations of the trusted certificates used to build the chain can be specified by the **-CAfile**, **-CApath** or **-CAstore** options or they will be looked for in the standard OpenSSL certificates directory.

If the initial verify fails then the OCSP verify process halts with an error.

Otherwise the issuing CA certificate in the request is compared to the OCSP responder certificate: if there is a match then the OCSP verify succeeds.

Otherwise the OCSP responder certificate's CA is checked against the issuing CA certificate in the request. If there is a match and the OCSPSigning extended key usage is present in the OCSP responder certificate then the OCSP verify succeeds.

Otherwise, if **-no_explicit** is **not** set the root CA of the OCSP responders CA is checked to see if it is trusted for OCSP signing. If it is the OCSP verify succeeds.

If none of these checks is successful then the OCSP verify fails.

What this effectively means if that if the OCSP responder certificate is authorised directly by the CA it is issuing revocation information about (and it is correctly configured) then verification will succeed.

If the OCSP responder is a "global responder" which can give details about multiple CAs and has its own separate certificate chain then its root CA can be trusted for OCSP signing. For example:

```
openssl x509 -in ocspCA.pem -addtrust OCSPSigning -out trustedCA.pem
```

Alternatively the responder certificate itself can be explicitly trusted with the **-VAfile** option.

NOTES

As noted, most of the verify options are for testing or debugging purposes. Normally only the **-CApath**, **-CAfile**, **-CAstore** and (if the responder is a 'global VA') **-VAfile** options need to be used.

The OCSP server is only useful for test and demonstration purposes: it is not really usable as a full OCSP responder. It contains only a very simple HTTP request handling and can only handle the POST form of OCSP queries. It also handles requests serially meaning it cannot respond to new requests until it has processed the current one. The text index file format of revocation is also inefficient for large quantities of revocation data.

It is possible to run this command in responder mode via a CGI script using the **-reqin** and **-respout** options.

EXAMPLES

Create an OCSP request and write it to a file:

```
openssl ocsp -issuer issuer.pem -cert c1.pem -cert c2.pem -reqout req.der
```

Send a query to an OCSP responder with URL http://ocsp.myhost.com/ save the response to a file, print it out in text form, and verify the response:

```
openssl ocsp -issuer issuer.pem -cert c1.pem -cert c2.pem \
  -url http://ocsp.myhost.com/ -resp_text -respout resp.der
```

Read in an OCSP response and print out text form:

```
openssl ocsp -respin resp.der -text -noverify
```

OCSP server on port 8888 using a standard **ca** configuration, and a separate responder certificate. All requests and responses are printed to a file.

```
openssl ocsp -index demoCA/index.txt -port 8888 -rsigner rcert.pem -CA demoCA/cacert.pem
     -text -out log.txt
```

As above but exit after processing one request:

```
openssl ocsp -index demoCA/index.txt -port 8888 -rsigner rcert.pem -CA demoCA/cacert.pem
  -nrequest 1
```

Query status information using an internally generated request:

```
openssl ocsp -index demoCA/index.txt -rsigner rcert.pem -CA demoCA/cacert.pem
  -issuer demoCA/cacert.pem -serial 1
```

Query status information using request read from a file, and write the response to a second file.

```
openssl ocsp -index demoCA/index.txt -rsigner rcert.pem -CA demoCA/cacert.pem
  -reqin req.der -respout resp.der
```

HISTORY

The -no_alt_chains option was added in OpenSSL 1.1.0.

COPYRIGHT