NAME

openssl-ocsp, ocsp - Online Certificate Status Protocol utility

SYNOPSIS

openssl ocsp [**-help**] [**-out file**] [**-issuer file**] [**-cert file**] [**-serial n**] [**-signer file**] [**-signkey file**] [**-sign_other file**] [**-no_certs**] [**-req_text**] [**-resp_text**] [**-text**] [**-reqout file**] [**-respout file**] [**-reqin file**] [**-respin file**] [**-nonce**] [**-no_nonce**] [**-url URL**] [**-host host:port**] [**-header**] [**-path**] [**-CApath dir**] [**-CAfile file**] [**-no-CAfile**] [**-no-CApath**] [**-attime timestamp**] [**-check_ss_sig**] [**-crl_check**] [**-crl_check_all**] [**-explicit_policy**] [**-extended_crl**] [**-ignore_critical**] [**-inhibit_any**] [**-inhibit_map**] [**-no_check_time**] [**-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**] [**-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**] [**-index file**] [**-CA file**] [**-rsigner file**] [**-rkey file**] [**-rother file**] [**-resp_no_certs**] [**-nmin n**] [**-ndays n**] [**-resp_key_id**] [**-nrequest n**] [**-md5|-sha1|...**]

DESCRIPTION

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

The **ocsp** 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. The certificate specified in **filename** must be in PEM format. 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.

-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.

-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.

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

-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

-CAfile file, **-CApath pathname**
file or pathname containing trusted CA certificates. These are used to verify the signature on the OCSP response.

-no-CAfile
Do not load the trusted CA certificates from the default file location

-no-CApath
Do not load the trusted CA certificates from the default directory location

-attime, **-check_ss_sig**, **-crl_check**, **-crl_check_all**, **-explicit_policy**, **-extended_crl**, **-ignore_critical**, **-inhibit_any**, **-inhibit_map**, **-no_alt_chains**, **-no_check_time**, **-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**
Set different certificate verification options. See *verify(1)* manual page for details.

-verify_other file
file 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.

-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 containing explicitly trusted responder certificates. Equivalent to the **-verify_other** and **-trust_other** options.

-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.

-[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.

OCSP Server Options

-index indexfile
indexfile is a text index file in **ca** format containing certificate revocation information.

If the **index** option is specified the **ocsp** utility is in 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 **indexfile**.

-rsigner file
The certificate to sign OCSP responses with.

-rother file
Additional certificates to include in the OCSP response.

-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.

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

-port portnum
Port to listen for OCSP requests on. The port may also be specified using the **url** option.

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

-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** and **CApath** 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** 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 the **ocsp** application 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/ca
-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/ca
-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 options was first added to OpenSSL 1.1.0.

COPYRIGHT