Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) was created as an alternative to *certificate revocation lists* (CRLs). Similar to CRLs, OCSP enables a requesting party (eg, a web browser) to determine the revocation state of a certificate.

When a CA signs a certificate, they will typically include an OCSP server address (eg, `http://ocsp.example.com`) in the certificate. This is similar in function to `crlDistributionPoints` used for CRLs.

As an example, when a web browser is presented with a server certificate, it will send a query to the OCSP server address specified in the certificate. At this address, an OCSP responder listens to queries and responds with the revocation status of the certificate.

> **Note**
>
> It's recommended to use OCSP instead where possible, though realistically you will tend to only need OCSP for website certificates. Some web browsers have deprecated or removed support for CRLs.

Prepare the configuration file

To use OCSP, the CA must encode the OCSP server location into the certificates that it signs. Use the `authorityInfoAccess` option in the appropriate sections, which in our case means the `[ server_cert ]` section.

```
[ server_cert ]
# ... snipped ...
authorityInfoAccess = OCSP;URI:http://ocsp.example.com
```

Create the OCSP pair

The OCSP responder requires a cryptographic pair for signing the response that it sends to the requesting party. The OCSP cryptographic

pair must be signed by the same CA that signed the certificate being checked.

Create a private key and encrypt it with AES-256 encryption.

```
# cd /root/ca
# openssl genrsa -aes256 \
      -out intermediate/private/ocsp.example.com.key.pem 4096
```

Create a certificate signing request (CSR). The details should generally match those of the signing CA. The **Common Name**, however, must be a fully qualified domain name.

```
# cd /root/ca
# openssl req -config intermediate/openssl.cnf -new -sha256 \
      -key intermediate/private/ocsp.example.com.key.pem \
      -out intermediate/csr/ocsp.example.com.csr.pem

Enter pass phrase for intermediate.key.pem: secretpassword
You are about to be asked to enter information that will be incorporat
into your certificate request.
-----
Country Name (2 letter code) [XX]:GB
State or Province Name []:England
Locality Name []:
Organization Name []:Alice Ltd
Organizational Unit Name []:Alice Ltd Certificate Authority
Common Name []:ocsp.example.com
Email Address []:
```

Sign the CSR with the intermediate CA.

```
# openssl ca -config intermediate/openssl.cnf \
      -extensions ocsp -days 375 -notext -md sha256 \
      -in intermediate/csr/ocsp.example.com.csr.pem \
      -out intermediate/certs/ocsp.example.com.cert.pem
```

Verify that the certificate has the correct **X509v3 extensions**.

```
# openssl x509 -noout -text \
      -in intermediate/certs/ocsp.example.com.cert.pem

    X509v3 Key Usage: critical
        Digital Signature
```

```
    X509v3 Extended Key Usage: critical
        OCSP Signing
```

Revoke a certificate

The OpenSSL `ocsp` tool can act as an OCSP responder, but it's only intended for testing. Production ready OCSP responders exist, but those are beyond the scope of this guide.

Create a server certificate to test.

```
# cd /root/ca
# openssl genrsa -out intermediate/private/test.example.com.key.pem 20
# openssl req -config intermediate/openssl.cnf \
      -key intermediate/private/test.example.com.key.pem \
      -new -sha256 -out intermediate/csr/test.example.com.csr.pem
# openssl ca -config intermediate/openssl.cnf \
      -extensions server_cert -days 375 -notext -md sha256 \
      -in intermediate/csr/test.example.com.csr.pem \
      -out intermediate/certs/test.example.com.cert.pem
```

Run the OCSP responder on `localhost`. Rather than storing revocation status in a separate CRL file, the OCSP responder reads `index.txt` directly. The response is signed with the OCSP cryptographic pair (using the `-rkey` and `-rsigner` options).

```
# openssl ocsp -port 127.0.0.1:2560 -text -sha256 \
      -index intermediate/index.txt \
      -CA intermediate/certs/ca-chain.cert.pem \
      -rkey intermediate/private/ocsp.example.com.key.pem \
      -rsigner intermediate/certs/ocsp.example.com.cert.pem \
      -nrequest 1

Enter pass phrase for ocsp.example.com.key.pem: secretpassword
```

In another terminal, send a query to the OCSP responder. The `-cert` option specifies the certificate to query.

```
# openssl ocsp -CAfile intermediate/certs/ca-chain.cert.pem \
      -url http://127.0.0.1:2560 -resp_text \
      -issuer intermediate/certs/intermediate.cert.pem \
      -cert intermediate/certs/test.example.com.cert.pem
```

The start of the output shows:

- whether a successful response was received (`OCSP Response Status`)
- the identity of the responder (`Responder Id`)
- the revocation status of the certificate (`Cert Status`)

```
OCSP Response Data:
    OCSP Response Status: successful (0x0)
    Response Type: Basic OCSP Response
    Version: 1 (0x0)
    Responder Id: ... CN = ocsp.example.com
    Produced At: Apr 11 12:59:51 2015 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: E35979B6D0A973EBE8AEDED75D8C27D67D2A0334
      Issuer Key Hash: 69E8EC547F252360E5B6E77261F1D4B921D445E9
      Serial Number: 1003
    Cert Status: good
    This Update: Apr 11 12:59:51 2015 GMT
```

Revoke the certificate.

```
# openssl ca -config intermediate/openssl.cnf \
      -revoke intermediate/certs/test.example.com.cert.pem

Enter pass phrase for intermediate.key.pem: secretpassword
Revoking Certificate 1003.
Data Base Updated
```

As before, run the OCSP responder and on another terminal send a query. This time, the output shows `Cert Status: revoked` and a `Revocation Time`.

```
OCSP Response Data:
    OCSP Response Status: successful (0x0)
    Response Type: Basic OCSP Response
    Version: 1 (0x0)
    Responder Id: ... CN = ocsp.example.com
    Produced At: Apr 11 13:03:00 2015 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: E35979B6D0A973EBE8AEDED75D8C27D67D2A0334
      Issuer Key Hash: 69E8EC547F252360E5B6E77261F1D4B921D445E9
      Serial Number: 1003
```

```
    Cert Status: revoked
    Revocation Time: Apr 11 13:01:09 2015 GMT
    This Update: Apr 11 13:03:00 2015 GMT
```

Comments

← Previous

Next →

Version 1.0.4 — Last updated on 2015-12-09.