Delegated Path Discovery with OCSP

1. Delegated Path Discovery

The path validation logic defined by [RFC2459] requires certificate-processing systems to accumulate the set of certificates from which certificate chains may be constructed as well as revocation data for each such certificate. These data may originate from diverse sources. Commonly used technologies for retrieving this information include X.500, LDAP, HTTP, FTP and SMTP as well as proprietary methods. Delegating this acquisition process to a separate server greatly simplifies and reduces the size of public-key based credential validation systems or other relying party software. It may also be useful to such software to be able to select from among various trust paths in the event multiple paths exist. The Delegated Path Discovery (DPD) extension to OCSP addresses these needs.

The DPD extension to OCSP request applies to the requestExtensions syntax of the OCSP request as outlined below (prior knowledge of [RFC2560] is assumed):
OCSP REQUEST
------------
In the requestExtensions field of TBSRequest, one extension MUST have an OID of
id-pkix-ocsp-path-req and a value of RetryReference, where

RetryReference ::= OCTET STRING

The RetryReference enables a requestor to acquire the next of potentially
several valid paths known to the OCSP server based on a previous response. If
this field is omitted then the request is considered to be a "new" request and
the responder may return any path that meets the request criteria. If a client
does specify a RetryReference then the responder MUST NOT return any path that
was previously returned with that reference (i.e. the responder MUST either
return a different path meeting the request or an error).

A DPD response consists of the following information:

OCSP RESPONSE
-------------
In the responseBytes field of OCSPResponse, responseType MUST have a value of
id-pkix-ocsp-path-rsp and response MUST have a value of DPDOCSPResponse, where

DPDOCSPResponse ::= SEQUENCE OF PathResponse
-- one for each certificate included in the requestList field of TBSRequest

PathResponse ::= SEQUENCE {
  retryReference   BIT STRING,
  certificates     SEQUENCE OF Certificate,
  crls             SEQUENCE SIZE (1..MAX) OF CertificateList OPTIONAL,
  ocspReps         SEQUENCE SIZE (1..MAX) OF OCSPResponse OPTIONAL
}

The sequence of certificates MUST form a potentially valid certification path,
in order, from end-entity certificate (element 0 of the sequence), up to and
including a "final" CA certificate, (which need not, but MAY be self-certified).

The RetryReference SHOULD uniquely refer to all path validation data (including
the data in the current response) that has been returned to the requester with
respect to this request.

The responder MAY also include a set of CRLs and/or OCSP responses which, if
included, SHOULD relate to the certificates in the set of certificates.

2. Conformance Requirements

An OCSP server claiming compliance to this specification SHALL:

1. Upon receipt of an authorized path discovery request, where possible, deliver
to the requestor a collection of certificates and optionally CRLs and other OCSP
responses that may be used to validate a certificate according to [RFC2459];

Myers et. al.
2. Either establish a stateful association enabling a requestor to serially ask for the next path via the retry option, to the extent that multiple validation paths exist and the receiving OCSP server is aware of these paths or respond with a noStateMaintained error to all retry requests if the server does not maintain state; and

3. In the event that the server is stateful and a prior response was the last path known to the responder, respond to subsequent retry requests with a noMoreData value in OSCPResponseStatus.

Requestors and responders SHALL at a minimum support the issuerSerial identification form of the ReqCert syntax of OCSP. Other identification forms MAY be supported according to local needs.

3. Security Considerations

A responder that only supports this service need not be trusted by a client for certificate status since it only supplies data that is signed by CAs. However, the client is trusting the responder to make an "honest effort" to find a path (or an additional path, if more than one exist). Since the client is presumably using the certificates for some important function, denial-of-service attacks on the responder are still potentially very serious and implementers should take steps to ensure the robustness of their implementations.

MORE TBD

4. References


5. Author’s Addresses

Michael Myers
VeriSign, Inc.
mmyers@verisign.com

Stephen Farrell
Baltimore Technologies
stephen.farrell@baltimore.ie

Carlisle Adams
Entrust Technologies
cadams@entrust.com
Appendix A : Collected Syntax

PathDiscovery DEFINITIONS EXPLICIT TAGS ::= 
{iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) pkix(7)
X -- TBS -- }

BEGIN

IMPORTS

-- PKIX
Certificate, CertificateList
FROM PKIX1Explicit88 {iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5)
pkix(7) id-mod(0) id-pkix1-explicit-88(1)}

-- OCSP
id-pkix-ocsp
FROM OCSP {iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5)
pkix(7) X -- TBD -- };

-- Delegated Path Discovery request
id-pkix-ocsp-path-req OBJECT IDENTIFIER ::= { id-pkix-ocsp X }

-- RetryReference ::= OCTET STRING --return next path, if one exists

-- Delegated Path Discovery response
id-pkix-ocsp-path-rsp OBJECT IDENTIFIER ::= { id-pkix-ocsp X }

DPDResponse ::= SEQUENCE {
    ref RetryReference,
certs SEQUENCE OF Certificate,
crls [0] SEQUENCE OF CertificateList OPTIONAL,
otherResps SEQUENCE OF OCSPResponse OPTIONAL
}

END