Network Working Group
Request for Comments: 3379
Category: Informational

D. Pinkas
Bull
R. Housley
RSA Laboratories

September 2002

Delegated Path Validation and Delegated Path Discovery Protocol Requirements

Status of this Memo

Copyright Notice

Abstract

This document specifies the requirements for Delegated Path Validation (DPV) and Delegated Path Discovery (DPD) for Public Key Certificates. It also specifies the requirements for DPV and DPD policy management.

1. Introduction

This document specifies the requirements for Delegated Path Validation (DPV) and Delegated Path Discovery (DPD) for Public Key Certificates, using two main request/response pairs.

Delegated processing provides two primary services: DPV and DPD. Some clients require a server to perform certification path validation and have no need for data acquisition, while some other clients require only path discovery in support of local path validation.

The DPV request/response pair, can be used to fully delegate path validation processing to an DPV server, according to a set of rules, called a validation policy.

The DPD request/response pair can be used to obtain from a DPD server all the information needed (e.g., the end-entity certificate, the CA certificates, full CRLs, delta-CRLs, OCSP responses) to locally validate a certificate. The DPD server uses a set of rules, called a path discovery policy, to determine which information to return.

A third request/response pair allows clients to obtain references for the policies supported by a DPV or DPD server.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document (in uppercase, as shown) are to be interpreted as described in [RFC2119].

2. Rationale and Benefits for DPV (Delegated Path Validation)

DPV allows a server to perform a real time certificate validation for a validation time T, where T may be the current time or a time in the recent past.

In order to validate a certificate, a chain of multiple certificates, called a certification path, may be needed, comprising a certificate of the public key owner (the end entity) signed by one CA, and zero or more additional certificates of CAs signed by other CAs.

Offloading path validation to a server may be required by a client that lacks the processing, and/or communication capabilities to fetch the necessary certificates and revocation information, perform certification path construction, and perform local path validation.

In constrained execution environments, such as telephones and PDAs, memory and processing limitations may preclude local implementation of complete, PKIX-compliant certification path validation [PKIX-1].

In applications where minimum latency is critical, delegating validation to a trusted server can offer significant advantages. The time required to send the target certificate to the validation server, receive the response, and authenticate the response, can be considerably less than the time required for the client to perform certification path discovery and validation. Even if a certification path were readily available to the client, the processing time associated with signature verification for each certificate in the path might (especially when validating very long paths or using a limited processor) be greater than the delay associated with use of a validation server.

Another motivation for offloading path validation is that it allows validation against management-defined validation policies in a consistent fashion across an enterprise. Clients that are able to do their own path validation may rely on a trusted server to do path validation if centralized management of validation policies is needed, or the clients rely on a trusted server to maintain centralized records of such activities.

When a client uses this service, it inherently trusts the server as much as it would its own path validation software (if it contained such software). Clients can direct the server to perform path validation in accordance with a particular validation policy.

3. Rationale and Benefits for DPD (Delegated Path Discovery)

DPD is valuable for clients that do much of the PKI processing themselves and simply want a server to collect information for them. The server is trusted to return the most current information that is available to it (which may not be the most current information that has been issued). The client will ultimately perform certification path validation.

A client that performs path validation for itself may get benefit in several ways from using a server to acquire certificates, CRLs, and OCSP responses [OCSP] as inputs to the validation process. In this context, the client is relying on the server to interact with repositories to acquire the data that the client would otherwise have to acquire using LDAP, HTTP, FTP [LDAP, FTP&HTTP] or another repository access protocol. Since these data items are digitally signed, the client need not trust the server any more than the client would trust the repositories.

DPD provides several benefits. For example, a single query to a server can replace multiple repository queries, and caching by the server can reduce latency. Another benefit to the client system is that it need not incorporate a diverse set of software to interact with various forms of repositories, perhaps via different protocols, nor to perform the graph processing necessary to discover certification paths, separate from making the queries to acquire path validation data.

4. Delegated Path Validation Protocol Requirements

4.1. Basic Protocol

The Delegated Path Validation (DPV) protocol allows a server to validate one or more public key certificates on behalf of a client according to a validation policy.

If the DPV server does not support the client requested validation policy, then the DPV server MUST return an error.

If the DPV request does not specify a validation policy, the server response MUST indicate the validation policy that was used.

Policy definitions can be quite long and complex, and some policies may allow for the setting of a few parameters (such as root self-signed certificates). The protocol MUST allow the client to include these policy dependent parameters in the DPV request; however, it is expected that most clients will simply reference a validation policy for a given application or accept the DPV server's default validation policy.

The client can request that the server determines the certificate validity at a time other than the current time. The DPV server MUST obtain revocation status information for the validation time in the client request.

In order to obtain the revocation status information of any certificate from the certification path, the DPV server might use, in accordance with the validation policy, different sources of revocation information. For example, a combination of OCSP responses, CRLs, and delta CRLs could be used. Alternatively, a response from another DPV server could be used.

If the revocation status information for the requested validation time is unavailable, then the DPV server MUST return a status indicating that the certificate is invalid. Additional information about the reason for invalidity MAY also be provided.

The certificate to be validated MUST either be directly provided in the request or unambiguously referenced, such as the CA distinguished name, certificate serial number, and the hash of the certificate, like ESSCertID as defined in [ESS] or OtherSigningCertificate as defined in [ES-F].

The DPV client MUST be able to provide to the validation server, associated with each certificate to be validated, useful certificates, as well as useful revocation information. Revocation information includes OCSP responses, CRLs, and delta CRLs. As an example, an S/MIME message might include such information, and the client can simply copy that information into the DPV request.

The DPV server MUST have the certificate to be validated. When the certificate is not provided in the request, the server MUST obtain the certificate and then verify that the certificate is indeed the one being unambiguous referenced by the client. The DPV server MUST include either the certificate or an unambiguous reference to the certificate (in case of a CA key compromise) in the DPV response.

The DPV response MUST indicate one of the following status alternatives:

1) the certificate is valid according to the validation policy.

2) the certificate is not valid according to the validation policy.

3) the validity of the certificate is unknown according to the validation policy.

4) the validity could not be determined due to an error.

When the certificate is not valid according to the validation policy, then the reason MUST also be indicated. Invalidity reasons include:

a) the DPV server cannot determine the validity of the certificate because a certification path cannot be constructed.

b) the DPV server successfully constructed a certification path, but it was not valid according to the validation algorithm in [PKIX-1].

c) the certificate is not valid at this time. If another request could be made later on, the certificate could possibly be determined as valid. This condition may occur before a certificate validity period has begun or while a certificate is suspended.

The protocol MUST prevent replay attacks, and the replay prevention mechanism employed by the protocol MUST NOT rely on synchronized clocks.

The DPV request MUST allow the client to request that the server include in its response additional information which will allow relying parties not trusting the DPV server to be confident that the certificate validation has correctly been performed. Such information may (not necessarily exclusively) consist of a certification path, revocation status information from authorized CRL issuers or authorized OCSP responders, revocation status information from CRL issuers or OCSP responders trusted under the validation

policy, time-stamp tokens from TSAs responders trusted under the validation policy, or a DPV response from a DPV server that is trusted under the validation policy. When the certificate is valid according to the validation policy, the server MUST, upon request, include that information in the response. However, the server MAY omit that information when the certificate is invalid or when it cannot determine the validity.

The DPV server MUST be able, upon request, copy a text field provided by the client into the DPV response. As an example, this field may relate to the nature or reason for the DPV query.

The DPV response MUST be bound to the DPV request so that the client can be sure that all the parameters from the request have been taken into consideration by the DPV server to build the response. This can be accomplished by including a one-way hash of the request in the response.

In some environments it may be necessary to present only a DPV response to another relying party without the corresponding request. In this case the response MUST be self contained. This can be accomplished by repeating only the important components from the request in the response.

For the client to be confident that the certificate validation was handled by the expected DPV server, the DPV response MUST be authenticated, unless an error is reported (such as a badly formatted request or unknown validation policy).

For the client to be able prove to a third party that trusts the same DPV server that the certificate validation was handled correctly, the DPV response MUST be digitally signed, unless an error is reported. The DPV server’s certificate MUST authenticate the DPV server.

The DPV server MAY require client authentication, therefore, the DPV request MUST be able to be authenticated.

When the DPV request is authenticated, the client SHOULD be able to include a client identifier in the request for the DPV server to copy into the response. Mechanisms for matching this identifier with the authenticated identity depends on local DPV server conditions and/or the validation policy. The DPV server MAY choose to blindly copy the identifier, omit the identifier, or return an error response.

There are no specific confidentiality requirements within this application layer protocol. However, when confidentiality is needed, it can be achieved with a lower-layer security protocol.

4.2. Relaying, Re-direction and Multicasting

In some network environments, especially ones that include firewalls, a DPV server might not be able to obtain all of the information that it needs to process a request. However, the DPV server might be configured to use the services of one or more other DPV servers to fulfill all requests. In such cases, the client is unaware that the queried DPV server is using the services of other DPV servers, and the client-queried DPV server acts as a DPV client to another DPV server. Unlike the original client, the DPV server is expected to have moderate computing and memory resources, enabling the use of relay, re-direct or multicasting mechanisms. The requirements in this section support DPV server-to-DPV server exchanges without imposing them on DPV client-to-DPV server exchanges.

Protocols designed to satisfy these requirements MAY include optional fields and/or extensions to support relaying, re-direction or multicasting. However, DPV clients are not expected to support relay, re-direct or multicast. If the protocol supports such features, the protocol MUST include provisions for DPV clients and DPV servers that do not support such features, allowing them to conform to the basic set of requirements.

- When a server supports a relay mechanism, a mechanism to detect loops or repetition MUST be provided.

- When a protocol provides the capability for a DPV server to re-direct a request to another DPV server (that is, the protocol chooses to provide a referral mechanism), a mechanism to provide information to be used for the re-direction SHOULD be supported. If such re-direction information is sent back to clients, then the protocol MUST allow conforming clients to ignore it.

- Optional parameters in the protocol request and/or response MAY be provide support for relaying, re-direction or multicasting. DPV clients that ignore any such optional parameters MUST be able to use the DPV service. DPV servers that ignore any such optional parameters MUST still be able to offer the DPV service, although they might not be able to overcome the limitations imposed by the network topology. In this way, protocol implementers do not need to understand the syntax or semantics of any such optional parameters.

5. Delegated Path Discovery Protocol Requirements

The Delegated Path Discovery (DPD) protocol allows the client to use a single request to collect at one time from a single server the data elements available at the current time that might be collected using

different protocols (such as LDAP, HTTP, FTP, or OCSP) or by querying multiple servers, to locally validate a public key certificate according to a single path discovery policy. The returned information can be used to locally validate one or more certificates for the current time.

Clients MUST be able to specify whether they want, in addition to the certification path, the revocation information associated with the path, for the end-entity certificate, for the CA certificates, or for both.

If the DPD server does not support the client requested path discovery policy, the DPD server MUST return an error. Some forms of path discovery policy can be simple. In that case it is acceptable to pass the parameters from the path discovery policy with each individual request. For example, the client might provide a set of trust anchors and separate revocation status conditions for the end-entity certificate and for the other certificates. The DPD request MUST allow more elaborated path discovery policies to be referenced. However, it is expected that most of the time clients will only be aware of the referenced path discovery policy for a given application.

The DPD server response includes zero, one, or several certification paths. Each path consists of a sequence of certificates, starting with the certificate to be validated and ending with a trust anchor. If the trust anchor is a self-signed certificate, that self-signed certificate MUST NOT be included. In addition, if requested, the revocation information associated with each certificate in the path MUST also be returned.

By default, the DPD server MUST return a single certification path for each end-entity certificate in the DPD request. However, the returned path may need to match some additional local criteria known only to the client. For example, the client might require the presence of a particular certificate extension or a particular name form. Therefore, the DPD client MUST have a means of obtaining more than one certification path for each end-entity certificate in the DPD request. At the same time, the mechanism for obtaining additional certification paths MUST NOT impose protocol state on the DPD server. Avoiding the maintenance of state information associated with previous requests minimizes potential denial of service attacks and other problems associated with server crashes.

Path discovery MUST be performed according to the path discovery policy. The DPD response MUST indicate one of the following status alternatives:

1) one or more certification paths was found according to the path discovery policy, with all of the requested revocation information present.

2) one or more certification paths was found according to the path discovery policy, with a subset of the requested revocation information present.

3) one or more certification paths was found according to the path discovery policy, with none of the requested revocation information present.

4) no certification path was found according to the path discovery policy.

5) path construction could not be performed due to an error.

When no errors are detected, the information that is returned consists of one or more certification paths and, if requested, its associated revocation status information for each certificate in the path.

For the client to be confident that all of the elements from the response originate from the expected DPD server, an authenticated response MAY be required. For example, the server might sign the response or data authentication might also be achieved using a lower-layer security protocol.

The DPD server MAY require client authentication, allowing the DPD request MUST to be authenticated.

There are no specific confidentiality requirement within the application layer protocol. However, when confidentiality is needed, it can be achieved with a lower-layer security protocol.

6. DPV and DPD Policy Query

Using a separate request/response pair, the DPV or DPD client MUST be able to obtain references for the default policy or for all of the policies supported by the server. The response can include references to previously defined policies or to a priori known policies.

7. Validation Policy

A validation policy is a set of rules against which the validation of the certificate is performed.

A validation policy MAY include several trust anchors. A trust anchor is defined as one public key, a CA name, and a validity time interval; a trust anchor optionally includes additional constraints. The use of a self-signed certificate is one way to specify the public key to be used, the issuer name, and the validity period of the public key.

Additional constraints for each trust anchor MAY be defined. These constraints might include a set of certification policy constraints or a set of naming constraints. These constraints MAY also be included in self-signed certificates.

Additional conditions that apply to the certificates in the path MAY also be specified in the validation policy. For example, specific values could be provided for the inputs to the certification path validation algorithm in [PKIX-1], such as user-initial-policy-set, initial-policy-mapping-inhibit, initial-explicit-policy, or initial-any-policy-inhibit.

Additional conditions that apply to the end-entity certificate MAY also be specified in the validation policy. For example, a specific name form might be required.

In order to succeed, one valid certification path (none of the certificates in the path are expired or revoked) MUST be found between an end-entity certificate and a trust anchor and all constraints that apply to the certification path MUST be verified.

7.1. Components for a Validation Policy

A validation policy is built from three components:

1. Certification path requirements,

2. Revocation requirements, and

3. End-entity certificate specific requirements.

Note: [ES-P] defines ASN.1 data elements that may be useful while defining the components of a validation policy.

7.2. Certificate Path Requirements

The path requirements identify a sequence of trust anchors used to start certification path processing and initial conditions for certification path validation as defined in [PKIX-1].

7.3. Revocation Requirements

Revocation information might be obtained through CRLs, delta CRLs or OCSP responses. Certificate revocation requirements are specified in terms of checks required on the end-entity certificate and CA certificates.

Revocation requirements for the end-entity certificate may not be the same as the requirements for the CA certificates. For example, an OCSP response may be needed for the end-entity certificate while CRLs may be sufficient for the CA certificates.

The validation policy MUST specify the source of revocation information:

- full CRLs (or full Authority Revocation Lists) have to be collected.
- OCSP responses, using [OCSP], have to be collected.
- delta CRLs and the relevant associated full CRLs (or full Authority Revocation Lists) are to be collected.
- any available revocation information has to be collected.
- no revocation information need be collected.

7.4. End-entity Certificate Specific Requirements

The validation policy might require the end-entity certificate to contain specific extensions with specific types or values (it does not matter whether they are critical or non-critical). For example, the validation policy might require an end-entity certificate that contains an electronic mail address (either in the rfc822 subject alt name or in the emailAddress naming attribute in the subject name).

8. Path Discovery Policy

A path discovery policy is a set of rules against which the discovery of a certification path is performed. A path discovery policy is a subset of a validation policy. A path discovery policy MAY either be a reference to a validation policy or contain only some major elements from a validation policy, such as the trust anchors.

Since the DPD client is "PKI aware", it can locally apply additional selection criteria to the certification paths returned by the server. Thus, a simpler policy can be defined and used for path discovery.

8.1. Components for a Path Discovery Policy

The path discovery policy includes certification path requirements, revocation requirements, and end-entity certificate specific requirements. These requirements are the same as those specified in sections 7.2, 7.3, and 7.4, respectively.

9. Security Considerations

A DPV client must trust a DPV server to provide the correct answer. However, this does not mean that all DPV clients will trust the same DPV servers. While a positive answer might be sufficient for one DPV client, that same positive answer will not necessarily convince another DPV client.

Other clients may trust their own DPV servers, or they might perform certification path validation themselves. DPV clients operating under an organizational validation policy must ensure that each of the DPV servers they trust is operating under that organizational validation policy.

When no policy reference is present in the DPV request, the DPV client ought to verify that the policy selected by the DPV server is appropriate.

The revocation status information is obtained for the validation time. In case of a digital signature, it is not necessarily identical to the time when the private key was used. The validation time ought to be adjusted by the DPV client to compensate for:

1) time for the end-entity to realize that its private key has been or could possibly be compromised, and/or

2) time for the end-entity to report the key compromise, and/or

3) time for the revocation authority to process the revocation request from the end-entity, and/or

4) time for the revocation authority to update and distribute the revocation status information.

10. Acknowledgments

These requirements have been refined after some valuable inputs from Trevor Freeman, Paul Hoffman, Ambarish Malpani, Mike Myers, Tim Polk, and Peter Sylvester.

11. References

11.1. Normative References

[PKIX-1] Housley, R., Ford, W., Polk, W. and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 3280, April 2002.

[OCSP] Myers, M., Ankney, R., Malpani, A., Galperin, S. and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.

11.2. Informative References

[ES-F] Pinkas, D., Ross, J. and N. Pope, "Electronic Signature Formats for long term electronic signatures", RFC 3126, September 2001.

[ES-P] Pinkas, D., Ross, J. and N. Pope, "Electronic Signature Policies", RFC 3125, September 2001.

[ESS] Hoffman, P., "Enhanced Security Services for S/MIME", RFC 2634, June 1999.

[ISO-X509] ISO/IEC 9594-8/ITU-T Recommendation X.509, "Information Technology - Open Systems Interconnection: The Directory: Authentication Framework," 1997 edition.

[FTP&HTTP] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure. Operational Protocols: FTP and HTTP", RFC 2585, May 1999.

[LDAP] Boeyen, S., Howes, T. and P. Richard, "Internet X.509 Public Key Infrastructure Operational Protocols LDAPv2", RFC 2559, April 1999.

12. Authors' Addresses

Denis Pinkas
Bull
Rue Jean-Jaures - BP 68
78340 Les Clayes-sous-Bois
FRANCE

EMail: Denis.Pinkas@bull.net

Russell Housley
RSA Laboratories
918 Spring Knoll Drive
Herndon, VA 20170
USA

EMail: rhousley@rsasecurity.com

13. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.