Network Working Group
Request for Comments: 5697
Category: Experimental

S. Farrell
Trinity College Dublin

November 2009

Other Certificates Extension

Abstract

Some applications that associate state information with public key certificates can benefit from a way to link together a set of certificates that belong to the same end entity and that can safely be considered equivalent to one another for the purposes of referencing that application-state information. This memo defines a certificate extension that allows applications to establish the required linkage without introducing a new application protocol data unit.

Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Table of Contents

1. Introduction

RFC 5280 [RFC5280] defines a profile for the use of public key certificates for Internet applications. If an application associates application-state information with a public key certificate, then that association may be disrupted if the end entity changes its public key certificate. Such disruption can occur due to renewals or if the end entity changes its certificate issuer. Similarly, if the end entity is actually a distributed system, where each instance has a different private key, then the relying party (RP) has no way to associate the different public key certificates with the relevant application-state information.

For example, assume a web browser retains state information (perhaps passwords) about a web site, indexed (possibly indirectly) via values contained in the web server's public key certificate (perhaps a DNS name). When the web server certificate expires and a new certificate is acquired (perhaps with a different DNS name), then the browser cannot safely map the new certificate to the relevant state information.

This memo defines a new public key certificate extension that supports such linkage, allowing the certificate issuer to attest that the end entity that holds the private key for the certificate in question also holds other private keys corresponding to other identified certificates.

Other than the issuer asserting that the set of certificates belongs to the same end entity for use with the same application, the fine detail of the semantics of the linkage of certificates is not defined here, since that is a matter for application developers and the operators of certification authorities (CAs). In particular, we do not define how a CA can validate that the same end entity is the holder of the various private keys, nor how the application should

make use of this information. Nor do we define what kinds of state information may be shared.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. A Use Case

Public key certificates expire, typically about a year after they are created. Some applications might need to know that the same entity is the subject of the current certificate and a previously used certificate.

For example, if a web server certificate expires, it could be useful for a web browser to know that the server currently presenting a certificate in a Transport Layer Security (TLS) [RFC5246] handshake represents the same web server that previously presented a certificate. This could be used, for example, to allow the browser to automatically fill in form fields for the server in question, even if the server certificate has been replaced. While the same effect can be achieved based on the use of the same issuer and subject fields in a certificate, there could be security issues involved in such comparisons, e.g., if the subject name includes a DNS name and the ownership of that DNS domain has changed.

The use of the new extension provides a way for the CA to signal to the application that the same end entity is involved, regardless of name changes. The new extension could also allow the web site operator to more easily change the CA when replacing its certificate.

3. Other Certificates Extension

This section defines the syntax for the other certificates extension.

The new extension is simply a list of references to the linked certificates. The references make use of the SCVPCertID structure from the Server-Based Certificate Validation Protocol (SCVP) [RFC5055], which contains a hash over the relevant certificate and the certificate's issuer and serial number.

When this extension is present, the CA is asserting that the same end entity is the subject of the relevant certificates.

This extension MUST NOT be marked critical.

```
id-pe-otherCerts OBJECT IDENTIFIER ::= { id-pe 19 }
```

```
OtherCertificates ::= SEQUENCE OF SCVPCertID
```

CAs MUST only issue certificates containing this extension where the links created are such that the relevant consumers of the certificates can safely make use of those links. This will typically be the case where the certificates are only used by a single application. CAs MUST NOT issue certificates that link to certificates issued for a different purpose, for example, a CA SHOULD NOT link a web server certificate to a VPN gateway certificate (unless those can be the same, which might occur for some embedded devices). The purpose for which the certificate is intended may be determined by certificate policy or other means (e.g., extended key usage object identifiers) that are out of the scope of this specification.

CAs MUST NOT issue certificates containing this extension unless they have validated that the end entity is the holder of all of the relevant private keys.

Applications MUST validate certificates according to the rules specified in RFC 5280 [RFC5280] and MUST NOT assume that because certificates are linked that they are therefore valid. This means, of course, that both certificates must chain up to some local trust point(s).

If an application imposes further checks on certificate validity (e.g., as is done in RFC 2818 [RFC2818] for web server certificates), then both certificates MUST be valid according to those application-specific rules.

It is not required that two linked certificates be simultaneously valid. For example, an application can validate certificate1 and cache that information. When the application is subsequently presented with certificate2 (linked back to certificate1), if it considers the cached information about certificate1 trustworthy, then it can validate certificate2 and use the linkage to associate certificate2 with the relevant application-state information (just as it would have done had certificate1 been re-presented). As a second example, if certificate1 has expired but would otherwise be valid, then the linkage from certificate2 can also be used once certificate2 has been validated.

If the application checks certificate status for the certificates in question, and any of the certificates concerned has been revoked, then the linkage MUST NOT be used.

Note that there are no constraints on the contents of the certificate to which the link points. The consequence is that the CA issuing the new certificate can link back to legacy certificates of all kinds, once the relevant RP supports this extension.

This extension MUST only be used in end-entity certificates, that is, it MUST NOT be used in CA certificates or other similar certificates. Since CA certificates are only used for certificate validation and this extension has no effect on the validation procedure, this extension would generally be meaningless in a CA certificate. In addition, it may be wise to gain some deployment experience with this extension before using it for more security-sensitive certificates, like CA certificates.

4. Another Approach Using Permanent Identifiers

RFC 4043 [RFC4043] defines a new name form (a "Permanent Identifier" or PI) for public key certificates that supports similar functionality to the new extension defined here. If two certificates have the same PI and that PI form is globally unique, then the end entities involved can be considered to be the same.

The main difference between the PI and the other certificates extension is that (when more than one CA is involved) PI requires a globally unique identifier, whereas the other certificates extension only requires that the issuer of the new certificate be able to link back to the old certificate(s).

As a consequence, the other certificates extension can be deployed "reactively" to link certificates that may not match "ideal" application-naming requirements. If the old certificate did make use of PI, then presumably application-naming issues have already been handled, and then the new certificate can contain the same PI. In this latter case, there would be no need for the other certificates extension.

5. A Possible Optimisation

The SCVPCertID structure used here contains the issuer name for the CA of the linked certificate. It may happen that this issuer is also the issuer of the certificate containing the other certificates extension. If a new certificate were linked back to a number of old certificates from that same CA, then there would be considerable redundancy since there would be many copies of the same issuer name.

One suggestion raised was to have a convention where if the X.500 Name in the SCVPCertID is an "empty" DN (see RFC5280), then that would indicate that the same CA issued both the current and the

linked certificates. However, that scheme is not adopted in this version. A future, Standards Track version of this specification might adopt that optimisation.

6. Acknowledgements

The use case motivating this was contributed to the W3C web security context (WSC) working group by Tyler Close. See http://www.w3.org/2006/WSC/wiki/SafeWebFormEditor for details.

Denis Pinkas pointed out that the PI extension is an alternative to this one.

James Manger suggested the optimisation to reduce the number of copies of the issuer name.

7. Security Considerations

As stated above, relying parties MUST validate any certificates per the algorithm given in RFC 5280 [RFC5280] before making any use of those certificates.

Relying parties similarly MUST NOT assume that any other fields in the relevant certificates have common values. For example, linked certificates might have non-overlapping key usage extensions.

Since the issuer of the new certificate (or some superior CA) is trusted by the RP, and the RP has validated the new certificate, the RP is basically as reliant on the proper operation of that CA as always -- if the CA wished to "cheat" on the RP, the other certificates extension simply provides a new way to do that, but one that is equivalent to existing vulnerabilities. In many cases, such a bad CA could simply issue a new certificate that is identical in all respects (other than the key pair) and the RP would accept the identity contained in that new certificate.

However, if the issuer of the new certificate is limited in some way (e.g., via a name constraint in a superior CA certificate), and if the old certificate doesn't match those limitations (e.g., the subject of the old certificate doesn't fit under the name constraints of the issuer of the new certificate), then the new certificate could be linked back to an identity that doesn't meet the constraints intended to be imposed on the issuer of the new certificate. Applications for which this is an unacceptable risk SHOULD NOT make use of the other certificates extension.

Since the SCVPCertID structure includes a hash of the other certificate and hash algorithm weaknesses that produce collisions are becoming more of an issue, CAs and relying parties MUST ensure that currently acceptable hash functions are used. In particular, the default use of SHA-1 for SCVPCertID may or may not currently be considered acceptable. CAs might be wise to use SHA-256 instead, but will typically use whatever hash function they use as part of certificate signing.

In some application contexts, if the old certificate has expired (and perhaps any associated certificate revocation list (CRL) entries are no longer on the latest CRL), it may be unsafe to link the new and old certificates. Application developers SHOULD carefully consider whether to make use of the other certificates extension in such contexts.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5055] Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, "Server-Based Certificate Validation Protocol (SCVP)", RFC 5055, December 2007.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

8.2. Informative References

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC4043] Pinkas, D. and T. Gindin, "Internet X.509 Public Key Infrastructure Permanent Identifier", RFC 4043, May 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Appendix A. ASN.1 Module

PKIX OID registrations may be viewed at:
http://www.imc.org/ietf-pkix/pkix-oid.asn

```
   PKIXOtherCertsModule
        { iso(1) identified-organization(3) dod(6) internet(1)
          security(5) mechanisms(5) pkix(7) id-mod(0) 44 }

   DEFINITIONS EXPLICIT TAGS ::=

   BEGIN

   -- EXPORTS ALL

   IMPORTS

   -- From [RFC5055]
   SCVPCertID
   FROM SCVP { iso(1) identified-organization(3) dod(6) internet(1)
                  security(5) mechanisms(5) pkix(7) id-mod(0) 21 } ;

   -- The one and only new thing, a new certificate extension

   id-pe-otherCerts OBJECT IDENTIFIER ::=
             { iso(1) identified-organization(3) dod(6) internet(1)
                  security(5) mechanisms(5) pkix(7) id-pe(1) 19 }

   -- The value is a sequence of cert ids.
   OtherCertificates ::= SEQUENCE OF SCVPCertID

   END
```

Author's Address

Stephen Farrell
Trinity College Dublin
Department of Computer Science
Trinity College
Dublin,   2
Ireland

Phone: +353-1-896-2354
EMail: stephen.farrell@cs.tcd.ie