Network Working Group
Internet Draft
Intended Status: Standard Track

Expires: June 5, 2008

Sean Turner
IECA
Santosh Chokhani
Orion Security Solutions
December 5, 2007

Clearance and CA Clearance Constraints Certificate Extensions
draft-turner-caclearanceconstraints-00.txt

Status of this Memo

Copyright Notice

Abstract

This document defines the syntax and semantics for the Clearance and the Certification Authority (CA) Clearance Constraints X.509 certificate extensions. The Clearance certificate extension is used to indicate the clearance held by the subject. The CA Clearance Constraints certificate extension values in a Trust Anchor (TA) and

the CAs in a certification path constrain the effective Clearance of the subject of the last certificate in the certification path.

Table of Contents

1. Introduction

Organizations that have implemented a security policy can issue certificates that include an indication of the clearance values held by the subject. The Clearance certificate extension indicates the security policy, the clearance levels held by the subject, and additional authorization information held by the subject. This specification makes use of the ASN.1 syntax for clearance from [RFC3281].

Some organizations have multiple TAs and/or CAs, and these organizations may wish to indicate to relying parties which clearance values from a particular TA or CA should be accepted. For example, consider the security policies described in [RFC3114], where a security policy has been defined for Amoco with three security classification values (HIGHLY CONFIDENTIAL, CONFIDENTIAL, and GENERAL). To constrain a CA for just one security classification, the

CA Clearance Constraints certificate extension would be included in the CA's certificate.

Cross-certified domains can also make use of the CA Clearance Constraints certificate extension to indicate which clearance values should be acceptable to relying parties.

1.1. Terminology

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 [RFC2119].

1.2. ASN.1 Syntax Notation

All X.509 certificate [RFC3280] extensions are defined using ASN.1 [X.680, X.690].

2. Clearance Certificate Extension

The Clearance certificate extension in a certificate indicates the clearances held by the subject. It uses the clearance attribute syntax from Section 4.4.6 of [RFC3281] in the Subject Directory Attributes extension. The Clearance certificate extension MUST never be marked critical. It is only meaningful if at least one of the following key usage bits is set: digital signature, non-repudiation, key transport, or key agreement. A certificate MUST include either zero or one instance of the Clearance certificate extension.

The following object identifier identifies the Clearance certificate extension:

```
id-at-clearance OBJECT IDENTIFIER ::= { joint-iso-ccitt(2)
  ds(5) module(1) selected-attribute-types(5) clearance(55) }
```

The ASN.1 syntax for the Clearance certificate extension is as follows:

```
Clearance  ::=  SEQUENCE {
  policyId           [0] OBJECT IDENTIFIER,
  classList          [1] ClassList DEFAULT {unclassified},
```

```
    securityCategories [2] SET OF SecurityCategory OPTIONAL
  }
```



```
  ClassList  ::=  BIT STRING {
    unmarked       (0),
    unclassified   (1),
    restricted     (2),
    confidential   (3),
    secret         (4),
    topSecret      (5)
  }

  SecurityCategory ::= SEQUENCE {
    type      [0]  IMPLICIT OBJECT IDENTIFIER,
    value     [1]  ANY DEFINED BY type
  }
```

The fields in Clearance certificate extension take their meaning from Section 4.4.6 of [RFC3281], which is repeated here for convenience:

- policyId identifies the security policy to which the clearance relates. The policyId indicates the semantics of the classList and securityCategory fields.

- classlist identifies the security classifications. Six basic values are defined in bit positions 0 through 5 and more may be defined by an organizational security policy.

- securityCategories provides additional authorization information.

If a trust anchor's public key is used directly, then the Clearance associated with the trust anchor, if any, should be used as the effective clearance (also defined as effective-clearance for a certification path).

3. CA Clearance Constraints Certificate Extension

The CA Clearance Constraints certificate extension indicates to the relying party what clearances should be acceptable for the subject of the last certificate in the certification path containing the TA or the CA. It is only meaningful in trust anchor or CA certificates. A trust anchor or CA certificate MUST include either zero or one instance of the CA Clearance Constraints certificate extension. The CA Clearance Constraints certificate extension MAY be critical or non-critical.

Absence of this certificate extension in a CA certificate or in a TA indicates that clearance of the subject of the last certificate in the certification path containing the CA or the TA is not constrained by the respective CA or TA.

The following object identifier identifies the CA Clearance Constraints certificate extension:

```
id-ce-caClearanceConstraints OBJECT IDENTIFIER ::= { id-TBSL }
```

The ASN.1 syntax for the CA Clearance Constraints certificate extension is as follows:

```
CAClearanceConstraints ::= SEQUENCE SIZE (1..MAX) OF Clearance
```

The syntax for CA Clearance Constraints certificate extension contains Clearance values that the CA asserts. The sequence MUST NOT include more than one entry with the same policyId. This constraint is enforced during Clearance and CA Clearance Constraints Processing described below. If more than one entry with the same policyId is present in CAClearanceConstraints certificate extension, the certification path is rejected.

4. Clearance and CA Clearance Constraints Processing

CA Clearance Constraints certificate extension processing determines the effective clearance (henceforth called effective-clearance) for the end certificate. CA Clearance Constraints certificate extension in the TA and in each certificate up to but not including the end certificate in a certification path impact the effective-clearance. If there is more than one path to the end-entity certificate, each path is processed independently. The process involves two steps:

1) collecting the CA Clearance Constraints; and

2) using CA Clearance Constraints in the certification path and the Clearance in the end certificate to determine the effective-clearance for the subject of the end certificate.

Assuming a certification path consisting of n certificates, the effective-clearance for the subject of the end certificate is the intersection of Clearance in the subject certificate, CA Clearance Constraints, if present, in trust anchor and all CA Clearance Constraints present in intermediate certificates. Any effective-clearance calculation algorithm that performs this calculation and provides the same outcome as the one from the algorithm described herein is considered compliant with the requirements of this RFC.

When processing a certification path, CA Clearance Constraints are maintained in one state variable: permitted-clearances. When processing begins, permitted-clearances is initialized to the special value all-clearances if CA Clearance Constraints certificate

extension is not present in the trust anchor, otherwise this value is initialized to CA Clearance Constraints associated with the trust anchor. The permitted-clearances state variable is updated each time an intermediate certificate that contains a CA Clearance Constraints certificate extension in the path is processed.

When processing the end certificate, the value in the Clearance certificate extension in the end certificate is intersected with the permitted-clearances state variable.

The output of Clearance and CA Clearance Constraint certificate extensions processing is the effective-clearance, which could also be an empty list; and success or failure with reason code for failure.

4.1. Collecting Constraints

CA Clearance Constraints are collected from the trust anchor and the intermediate certificates in a certification path.

4.1.1. Certification Path Processing

When processing CA Clearance Constraints certificate extension for the purposes of validating Clearance in the end certificate, the

processing described in this section or an equivalent algorithm MUST be included in the certification path validation. The processing is presented as additions to the certification path validation algorithm described in section 6 of [RFC3280].

4.1.1.1. Inputs

Trust anchor information may include the CAClearanceConstraints structure to specify CA Clearance Constraints for the trust anchor. The trust anchor may be constrained or unconstrained.

4.1.1.2. Initialization

Examine the trust anchor and verify that it does not contain more than one instance of CAClearanceConstraints extension. If the trust anchor contains more than one instance of CAClearanceConstraints extension, set effective-clearance to an empty list, set error code to "multiple extension instances", and exit with failure.

Create a state variable named permitted-clearances. If the trust anchor contains a CAClearanceConstraints extension, then the initial value of permitted-clearances is the CAClearanceConstraints extension from the trust anchor.

Examine the permitted-clearances for the same Policy ID appearing more then once. If a policyID appears more than once in the permitted-clearance state variable, set effective-clearance to an empty list, set error code to "multiple instances of same clearance", and exit with failure..

If the trust anchor does not contain a CAClearanceConstraints extension, the permitted-clearances variable is assigned the special value all-clearances.

4.1.1.2.1. Basic Certificate Processing

If the certificate is the last certificate (i.e., certificate n), skip the steps listed in this section.

Examine the certificate and verify that it does not contain more than one instance of CAClearanceConstraints extension. If the certificate contains more than one instance of CAClearanceConstraints extension,

set effective-clearance to an empty list, set error code to "multiple extension instances", and exit with failure.

If the CAClearanceConstraints certificate extension is not present in the certificate, no action is taken, and the permitted-clearances value is unchanged.

If the CAClearanceConstraints certificate extension is present in the certificate, set the variable temp-clearances to CAClearanceConstraints certificate extension. Examine the temp-clearances for the same Policy ID appearing more then once. If a policyID appears more than once in the temp-clearances state variable, set effective-clearance to an empty list, set error code to "multiple instances of same clearance", and exit with failure.

If the CAClearanceConstraints certificate extension is present in the certificate and permitted-clearances contains the all-clearances special value, then assign permitted-clearances the value of the temp-clearances.

If the CAClearanceConstraints certificate extension is present in the certificate and permitted-clearances does not contain the all-clearances special value, take the intersection of temp-clearances and permitted-clearances by repeating the following steps for each clearance in the permitted-clearances state variable:

- If the policyID associated with the clearance is absent in the temp-clearances, delete the clearance structure associated with the policyID from the permitted-clearances state variable.

- If the policyID is present in the temp-clearances:

  -- For every classList bit, assign the classList bit a value of one (1) for the policyID in permitted-clearances state variable if the bit is one (1) in both the permitted-clearances state variable and the temp-clearances for that policyID; otherwise assign the bit a value of zero (0).

  -- If no bits are one (1) for the classList, delete the clearance structure associated with the policyID from the permitted-clearances state variable and skip the next step of processing securityCategories.

-- Calculate securityCategories intersection in accordance with guidelines associated with the security policy represented by the policyID.

4.1.1.2.2. Preparation for Certificate i+1

No additional action associated with the Clearance or CAClearanceConstraints certificate extensions is taken during this phase of certification path validation as described in section 6 of [RFC3280].

4.1.1.2.3. Wrap-up Procedure

To complete the processing, perform the following steps for the last certificate (i.e., certificate n).

Examine the certificate and verify that it does not contain more than one instance of Clearance extension. If the certificate contains more than one instance of Clearance extension, set effective-clearance to an empty list, set error code to "multiple extension instances", and exit with failure.

If the Clearance certificate extension is not present in the end certificate, set effective-clearance to an empty list and exit with success.

Set effective-clearance to the value from the Clearance certificate extension in the end certificate. Let us say policyID in effective-clearance is X.

If permitted-clearance is an empty list, set effective-clearance to an empty list and exit with success.

If the permitted-clearance has special value of all-clearances, exit with success.

If the policyID X in effective-clearance is absent from the permitted-clearance, set effective-clearance to an empty list and exit with success.

Assign those classList bits in effective-clearance a value of one (1)

that have a value of one (1) both in effective-clearance and in the clearance structure in permitted-clearance associated with policyID X. Assign all other classList bits in effective-clearance a value of zero (0).

If none of the classList bits have a value of one (1) in effective-clearance, set effective-clearance to an empty list and exit with success.

Set securityCategories in effective-clearance as an intersection of the securityCategories in the effective-clearance and securityCategories in the permitted-clearances for policyID X as defined by the policyID X.

Exit with Success

4.1.1.2.4. Outputs

If certification path validation processing succeeds, effective-clearance contains the effective clearance for the subject of the certification path. Processing also returns success or failure indication and reason for failure, if applicable.

5. Security Considerations

Certificate issuers must recognize that absence of the CAClearanceConstraints in a CA certificate means that in terms of the clearance, the subject CA is not constrained.

Absence of Clearance extension in a certificate means that the subject has not been assigned any clearance.

If there is no Clearance associated with a TA, it means that the TA has not been assigned any clearance.

If the local security policy considers the clearance held by a subject or those supported by a CA to be sensitive, then the Clearance or CA Clearance Constraints should only be included if the subject's and CA's certificate can be privacy protected. Also in

this case, distribution of trust anchors and associated CA Clearance Constraints extension or Clearance must also be privacy protected.

6. IANA Considerations

None. Please remove this section prior to publication as an RFC.

7. References

7.1. Normative References

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

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certification Revocation List (CRL) Profile", RFC 3280, April 2002.

[RFC3281] Farrell, S., and Housley, R., "An Internet Attribute Certificate Profile for Authorization", RFC 3281, April 2002.

[X.680] ITU-T Recommendation X.680: Information Technology - Abstract Syntax Notation One, 1997.

[X.690] ITU-T Recommendation X.690 Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 1997.

7.2. Informative References

[RFC3114] Nicolls, W., "Implementing Company Classification Policy with S/MIME Security Label", RFC3114, May 2002.

Appendix A. ASN.1 Module

This appendix provides the normative ASN.1 definitions for the structures described in this specification using ASN.1 as defined in X.680.

```
Clearance-CAClearanceConstraints93 { id-TBSL }

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS ALL --

IMPORTS

-- IMPORTS from RFC3281

Clearance
   FROM PKIXAttributeCertificate
   { iso(1) identified-organization(3) dod(6) internet(1)
     security(5) mechanisms(5) pkix(7) id-mod(0)
     id-mod-attribute-cert(12)
   }

EXTENSION
  FROM PKIX1Explicit93
   { iso(1) identified-organization(3) dod(6) internet(1)
     security(5) mechanisms(5) pkix(7) id-mod(0)
     id-pkix1-explicit-93(3)
}


;

-- Clearance certificate extension OID and syntax

clearance EXTENSION ::= {
  SYNTAX         Clearance
  IDENTIFIED BY  id-at-clearance
}

-- The following is a '93 version for clearance.
-- It is included for convenience.

-- id-at-clearance OBJECT IDENTIFIER ::=
--  { joint-iso-ccitt(2) ds(5) module(1) selected-attribute-types(5)
```

```
   --    clearance (55)
   -- }

   -- Clearance  ::=  SEQUENCE {
   --   policyId            [0] OBJECT IDENTIFIER,
   --   classList           [1] ClassList DEFAULT {unclassified},
   --   securityCategories  [2] SET OF SecurityCategory  OPTIONAL
   -- }

   -- ClassList  ::=  BIT STRING {
   --   unmarked      (0),
   --   unclassified  (1),
   --   restricted    (2),
   --   confidential  (3),
   --   secret        (4),
   --   topSecret     (5)
   -- }

   -- SECURITY-CATEGORY ::= TYPE-IDENTIFIER

   -- SecurityCategory ::= SEQUENCE {
   --   type  [0]
   --      IMPLICIT TYPE-IDENTIFIER.&id({SupportedSecurityCategories}),
   --   value [1]
   --      TYPE-IDENTIFIER.&Type({SupportedSecurityCategories}{@type})
   -- }

   -- CA Clearance Constraints certificate extension OID and syntax

   id-ce-caClearanceConstraints OBJECT IDENTIFIER ::= { id-TBSL }

   caClearanceConstraints EXTENSION ::= {
     SYNTAX         CAClearanceConstraints
     IDENTIFIED BY  id-ce-caClearanceConstraints
   }
   CAClearanceConstraints ::= SEQUENCE SIZE (1..MAX) OF Clearance

   END
```

Author's Addresses

Sean Turner

IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

EMail: turners@ieca.com

Santosh Chokhani
Orion Security Solutions, Inc.

Email: chokhani@orionsec.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).