Clearance Attribute and Authority Constraints Certificate Extension

draft-ietf-pkix-authorityclearanceconstraints-00.txt

IETF 73 – November 2008

Sean Turner

Santosh Chokhani

What's new?

- Published as WG ID.
- Section 1:
  - Added:
    - The clearance attribute can be in a public key or attribute certificate.
    - The authority clearance constraints extension can be in a CA's or AA's public key certificate.
  - Provided rationale for why clearance might be included in a public key and/or attribute certificate.

What's new?

- Section 2:
  - Required single value for clearance.
  - Aligned ASN.1 with module (uses '02 ASN.1 syntax).
- Section 5: Amended section so it's more like Section 4.
- Section 4/5/6: Added text to explain computation of security categories intersection.
- Section 7: Defined recommended security categories.

What's left?

- Agree that Authority Clearance Constraints should be optionally critical.
  - We proposed it's the CA's choice, but others felt that it should be critical.
- Come to closure on security categories intersection section.

Example: Notation

{{OID X, 111010, [OID A, 10110] [OID B 00110]}
{OID Y, 111010, [OID C, 10110] [OID D 00110]}}

- Each clearance is embedded in { }
- Different clearances represented using different color fonts (e.g., black and blue)
- OID X, Y, Z etc. represent security policy OID
- Classification represented by bits (e.g., 1001)
- Each security category is embedded in [ ]
- OID A, B, C, D etc. represent Type OID for security category
- Security category value is bit string a la recommended securityCategory value structure in the I-D (e.g.0101)

Example: Certificate Contents

- ACC in TA
  - {{OID X, 111010, [OID A, 10110] [OID B 00110]}
    {OID Y, 111010, [OID C, 10110] [OID D 00110]}}
  - Two clearances (X and Y)
  - Each clearance with two security categories (A&B, C&D)
- ACC in TA → CA
  - {{OID X, 110010, [OID A, 10100] [OID B 00110]}
    {OID Y, 100101, [OID C, 10110] [OID D 00110]}}
- ACC in CA → AA
  - {{OID X, 101010, [OID A, 00110] [OID B 00110]}
    {OID Y, 011010, [OID C, 10110] [OID D 00110]}}
- Clearance in AA → AC
  - {OID X, 011010, [OID A, 00110] [OID B 00110]}

Example: Processing TA

- Initial permitted-clearances
  - {{OID X, 111010, [OID A, 10110] [OID B 00110]} {OID Y, 111010, [OID C, 10110] [OID D 00110]}}
  - **Set the initial permitted-clearances value to the ACC in TA**

Example: Processing TA → CA

- permitted-clearances (Generic)
  - {{OID X, 110010, [OID B 00110]}
    {OID Y, 100000, [OID C, 10110] [OID D 00110]}}
  - **Classification bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the ACC remain set**
  - **securityCategory A deleted due to lack of exact match in values**
  - **securityCategory B, C, D remain intact due to exact match**
- permitted-clearances (Type OID Specific)
  - {{OID X, 110010, [OID A, 10100] [OID B 00110]}
    {OID Y, 100000, [OID C, 10110] [OID D 00110]}}
  - Classification bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the ACC remain set
  - **securityCategory A bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the ACC remain set**
  - securityCategory B, C, D remain intact due to exact match

Example: Processing CA → AA

- permitted-clearances (Generic)
  - {OID X, 100010, [OID B 00110]}
  - Classification bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the ACC remain set
  - **Clearance Y deleted due to no classification bits being set**
  - securityCategory B remains intact due to exact match
- permitted-clearances (Type OID Specific)
  - {OID X, 100010, [OID A, 00100] [OID B 00110]}
  - Classification bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the ACC remain set
  - Clearance Y deleted due to no classification bits being set
  - securityCategory A bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the ACC remain set
  - securityCategory B remains intact due to exact match

Example: Processing AA → AC

- effective-clearance (Generic)
  - Generic: {OID X, 000010, [OID B 00110]}
  - Classification bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the **clearance attribute** remain set
  - securityCategory B remains intact due to exact match
- effective-clearance (Type OID Specific)
  - {OID X, 000010, [OID A, 00100] [OID B 00110]}
  - Classification bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the clearance attribute remain set
  - securityCategory A bits that are set (i.e., 1) in both the state variable (i.e., permitted-clearances) and in the **clearance attribute** remain set
  - securityCategory B remains intact due to exact match

Questions

?