The PKIX UserGroupName GeneralName Type
draft-ietf-pkix-usergroup-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on July 12, 2001.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

A number of systems which understand X.509 client certificates have developed various ad-hoc mechanisms to map a certificate to a userid/group value which can then be used for access control. The mechanisms include peculiar name forms for the SubjectName field such as encoding the userid as a CommonName and the group as an OrganizationalUnit, or mapping the certificate against an entry in a directory system. This document describes an otherName extension of the GeneralName type which can be used in the SubjectAltName extension or IssuerAltName extension to directly encode userid and group information.
Table of Contents

1. Introduction ................................................. 3
2. Definition .................................................. 3
3. Usage ....................................................... 4
   3.1 Use Within a Leaf (End User) Certificate ................. 4
   3.2 Use Within an Intermediate (CA) Certificate .......... 4
4. Path Validation Considerations ............................ 5
   4.1 Trust Mappings ........................................ 5
   4.2 Domain Matching ....................................... 5
   4.3 Multiple UserGroupNames ............................... 6
5. Security Considerations ...................................... 6
6. Design Discussion ........................................... 7
   6.1 Attribute Certificate ................................ 7
   6.2 Certificate Extension ................................ 8
   6.3 otherName GeneralName type ............................ 8
       References ............................................. 8
6. Author’s Address ............................................ 8
A. ASN.1 Definitions .......................................... 8
   A.1 1988 Module ........................................... 9
   A.2 Notional ASN.1 Encoding ............................... 10
       Full Copyright Statement .............................. 11
1. Introduction

This document defines a proposed extension of the acceptable otherName instantiations of the GeneralName type within the Subject and IssuerAltName extensions described in [rfc2459]. It is applicable to the X.509 Public Key Infrastructure for the Internet family of standards. The definitions described herein extend and depend on those described in RFC 2459 which, in turn, defines the underlying certificate formats needed for a full implementation of this otherName definition.

This document describes a method of encoding Unix-style userid and group information directly within an X.509 certificate. Currently, a number of systems (e.g. web servers which accept or depend upon SSL client certificate authentication) use ad-hoc methods for either mapping from or encoding names within the X.509 SubjectName RDN (Relative Distinguished Name). For example, one system uses a CommonName element within the SubjectName to represent a userid and an OrganizationalUnit element to represent one or more groups. Another system stores a mapping from a particular certificate to a set of userid and group information within an LDAP database. Neither of the above approaches are standardized, nor are they substantially interoperable across many systems.

2. Definition

This name is defined as a form of otherName from the GeneralName structure in SubjectAltName. The basic definition of the UserGroupName type is:

```
id-on-userGroup   AttributeType ::= { id-on 2 }
UserGroupName ::=   SEQUENCE {
    domain            UTF8String,
    user              UTF8String,
    groups            SEQUENCE OF UTF8String OPTIONAL
}
```

The UserGroupName otherName consists of three fields:
- The domain field indicates the domain under which the other fields are evaluated. Although it is encoded as a UTF8String to permit future expansion, by convention this SHOULD be specified as a valid, ‘.’ (dot) separated tokens, DNS-style domain name. The domain is used to differentiate userids on various systems and within various organizations. For example, ‘smith’ on the host yoohoo.entera.com might be different than ‘smith’ on the host (or within the domain) bigbank.org. See Section 4 below for more information on the treatment of the domain field.
- The user field encodes the userid represented by this certificate...
within the domain specified. In general, this should be in the subset of UTF8 common to the target domain.

- The groups field is optional and encodes the groups the holder of this certificate is permitted access to within the specified domain.

3. Usage

3.1 Use Within a Leaf (End User) Certificate

For an end user certificate (i.e. a client certificate), the UserGroupName element is encoded within a SubjectAltName extension (SANE). The SANE SHOULD be marked as critical, and MUST be marked as critical if the the SubjectName field is empty.

The UserGroupName represents an identity for the certificate. In general, there SHOULD NOT be any other non-UserGroupName names within the SANE and the SubjectName field SHOULD be empty.

NOTE: Multiple UserGroupName elements are permitted with the SANE of a leaf certificate. Each is valid as an identity if and only if the acceptor can establish that the client certificate chains back to a CA certificate with a trust relationship for the domain indicated in that UserGroupName element. For any given acceptor context where the client certificate has multiple UserGroupNames, some of the UserGroupNames may be valid and some may not.

The selection of which of a number of valid multiple names is accepted by the server or other acceptor is an implementation decision. Possibilities include accepting the user with the most access, with the least access, prioritizing it based on an ordered list of domains or only accepting a single domain.

3.2 Use Within an Intermediate (CA) Certificate

For a CA certificate, one or more UserGroupName otherNames MAY be included within a SANE. The SANE MUST be marked as critical in that event. The inclusion of the UserGroupName element acts to restrict the set of groups which this CA (and its subsidiary CA’s) may certify. The userid field MAY be non-empty, but is ignored for most purposes. If a UserGroupName element is present in the SANE of a CA certificate, then the BasicConstraints extension MUST also be included and the BasicConstraints.ca flag MUST be set to true.

As per normal usage, the contents of the issuer’s SANE are generally copied to the IssuerAltName extension of an issued certificate as part of the certificate signature process. However, only the UserGroupName SANEs are consulted during UserGroupName path validation.
4. Path Validation Considerations

4.1 Trust Mappings

Each system (server, application) which accepts certificates with a UserGroupName element within the certificate’s SANE must establish one or more trust mappings between the specified domain tags and root or intermediate CA certificates. During path validation, the accepting system MUST verify that the offered certificate chains back to a root or intermediate CA that has a trust mapping which contains the certificate’s SubjectAltName UserGroup domain.

The specific mechanism for establishing or describing the trust mapping is outside of the scope of this document. However, it could be something as simple as a text file with the first column listing the domain and the second column listing a certificate fingerprint.

```
cacheflow.com  
entera.com  
```

Another possibility might be to identify the trusted certificate by subjectName and the subjectKeyIdentifier. This latter approach may be more useful as it allows expired roots to be easily superceded.

4.2 Domain Matching

As mentioned above, the domain field SHOULD be a DNS structured, dot separated string. To be valid a client certificate MUST chain back to a trusted certificate where the domain specified by the trust mapping is either equal to the domain of the client certificate or contains the domain of the client certificate. E.g.:

```
if trustDomain(CA) == userGroupName.domain
    then match
else if  tail (userGroupName.domain, length(trustDomain(CA)) + 1) ==
    concat (".", trustDomain)
    then match
else no match
```

depth (string, len) returns the last len characters of a string
depth (string, string) returns a concatenation of two strings

The addition of the ‘.’ (dot) to the trust domain in the algorithm ensures that an invalid match like "mystupiddomain.com" matching "stupiddomain.com" doesn’t happen. Also, as is normal for DNS style
names, matching is done without respect to case. Note that there may be multiple trust mappings for a single root and that all MUST be tried for the match.

4.3 Multiple UserGroupNames

CA certificates may contain a SANE with one or more UserGroupNames with a non-null domain and an empty or non-empty set of groups. If a CA certificate does have a such a SANE, it acts to restrict the set of groups that can be ‘certified’ by that CA certificate. The final set of groups output from the path validation processing of the UserGroupName elements is the MIN of the sets of all the group elements from each UserGroupName. The algorithm is:

for each UserGroupName (UGN) in the client certificate SANE
  set maxGroups to client’s UGN.groups
for each UGN in the CA certificate in the path
  if the CA’s UGN.domain contains or equals the clients UGN.domain
    then set maxGroups to MIN (maxGroups, CAs UGN.groups)
output {domain, maxGroups and userid}

For example, the client has a certificate issued by his host with a UserGroupName containing a user of ‘stjohns’, a group field of ['system', 'security', 'atg'], and a domain field of ‘atg.cacheflow.com’. It validly chains back to a root CA certificate that has a trust relationship for ‘cacheflow.com’. The CA which signed the client certificate has a SANE with one UserGroupName containing a domain of ‘cacheflow.com’ and groups of ['system', 'atg', 'admin'] and another UserGroupName containing a domain of ‘atg.cacheflow.com’ with groups of ['atg']. The certificate would be valid for the domain ‘atg.cacheflow.com’, the user ‘stjohns’ and the groups ['atg'].

5. Security Considerations

The use of the UserGroupName in most situations has roughly the same level of threat as that for any X.509 certificate and the Security Considerations text in [rfc2459] is applicable. However, the use of the groups element of the UserGroupName may have some unforeseen side effects in certain systems.

The user tags (along with the private keys of the certificates) are just credentials. They can be used to prove the identity of a client, but don’t of themselves grant a client access to data or other resources. An access control system is used to map credentials to rights. Revocation of access of a client to a resource is as simple as removing or changing the access control entry and doesn’t necessarily require the use of a Certificate
Revocation List (CRL) to revoke the certificate. Indeed, the revocation of the certificate may be undesirable as the removal of access may be only temporary.

However, the use of group tags is problematic in systems which do not consider Certificate Revocation Lists as part of their path validation processing. Since group access by definition covers a group of people, its difficult or impossible to remove access for a single member of the group without removing the user from the group. If the group information is encoded in the certificate, use of CRLs is pretty much mandatory.

One possible approach where groups are desired, but CRLs are not available is to not encode the group information within the certificate. Instead, do the user to group mapping as part of the access control processing (ala Unix). In that manner, a user could be deleted from a group simply by updating the user to group mappings on all accepting systems. Another related approach might be to have a user and group hotlist where you would add a user only if you wanted to remove them from a group that’s listed in the client certificate.

6. Design Discussion

There are many possible ways of accomplishing the goals of directly encoding user and group information within a certificate. This section provides some insight into the author’s reasoning to adopt the otherName approach.

6.1 Attribute Certificate

Attribute certificates (ACs) were offered by a couple of members of the PKIX community as one solution for this problem. The argument was made that attribute certificates were the approprate approach for things that looked like authorization tokens.

This approach was rejected for a couple of reasons. First, that ACs had no defined mechanism for encoding the user and group information. This otherName extension would still be required to provide the encoding. Second, ACs were neither in wide use, nor widely understood. Without this knowledge, there is a lack of software on both the issuer and acceptor ends which could deal with the ACs. Third, the author didn’t completely agree with the argument that userid and group credentials are authorization tokens. Userid and group assignments for most users tend to be fairly static. The author, for example, held the same userid and the same group assignments for over four years at his previous job. (Except, see Section 5 above.)
If ACs gain acceptability, there should be no bars to incorporating this otherName extension as one of the possible attributes allowed within an AC.

6.2 Certificate Extension

The approach of encoding the user and group information as a new CertificateExtension was initially attractive for a number of reasons. First, most software and toolkits had at least rudimentary support for new extensions. Second, the actual ASN1 encoding of the information was a bit less clumsy than if encoded as an otherName.

This option was also rejected when the author tried to consider the interactions between this extension, the Subject and IssuerNames, and the Subject and IssuerAltName extensions. Having a third place for possibly authoritative name information might make the rules for resolving which name should be used for various authorization activities a bit difficult.

6.3 otherName GeneralName type

Encoding the user and group information as an otherName appears to be the mostly right choice. It doesn’t add additional extensions which might further conflict with the SubjectName field and SubjectAltName extension. The usage is well within the appropriate field of use for the otherName type.

References


Author’s Address

Michael StJohns
CacheFlow
650 Almanor Ave
Sunnyvale, CA 94086
USA

Phone: +1-408-543-0476
EMail: stjohns@cacheflow.com

Appendix A. ASN.1 Definitions

N.B. Any assignments within this section should not be relied upon until and if this document is published either as an experimental
RFC or until it enters the standards track.

A.1 1988 Module

PKIXUsergroup (iso(1) identified-organization(3) dod(6)
internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
id-mod-user-group (20) )

DEFINITIONS EXPLICIT TAGS ::=  

BEGIN  

-- EXPORTS ALL --

IMPORTS  

id-pkix, AttributeType, UTF8String  
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))

-- Object Identifiers

-- Externally defined OIDs

-- Arc for other name forms
id-on OBJECT IDENTIFIER ::= { id-pkix 8 }

-- userGroupName

id-on-userGroup AttributeType ::= { id-on 2 }

UserGroupName ::= SEQUENCE {  
domain UTF8String,  
user UTF8String,  
groups SEQUENCE OF UTF8String OPTIONAL  
}

END
A.2 Notional ASN.1 Encoding

The following represents how a UserGroupName otherName would be encoded within a SubjectAltName Extension. The notation "'['" means to encode the enclosed items and return the octets.

```
[SEQUENCE                              -- Extension
  OID            id-ce-subjectAltName  -- extnID
  BOOLEAN        true                  -- critical
  OCTET STRING   -- extnValue
    -- Encode the following and provide as value for OCTET STRING
' [SEQUENCE                          -- GeneralNames
  [0                               -- otherName type tag
    [SEQUENCE                      -- otherName
      OID    id-on-userGroup       -- type-id
        [0
          [SEQUENCE                  -- UserGroupName
            UTF8String  'entera.com' -- domain
            UTF8String  'stjohns'    -- userid
          [SEQUENCE OF              -- groups
            UTF8String  'system'
            UTF8String  'atg'
          ]
        ]
    ]
  ]
]
]
```

