DataGRID

VOMS Credential Format

Describing Attributes Format

Document identifier:	**voms-credential**
EDMS id:	
Date:	January 14, 2004
Work package:	**EDG, DATATAG**
Partner(s):	**CERN, INFN**
Lead Partner:	**CERN, INFN**
Document status:	**WORKING DRAFT**
Author(s):	Ákos Frohner, Vincenzo Ciaschini
File:	**edg-voms-credential**

Abstract: Format of attributes and their encoding in Attribute Certificates and XACML pieces. The document also describes the “old” format.

CONTENTS

1. Fully Qualified Attribute Names

VOMS defines groups, roles and capabilities. Combinations of the names of these serve as attributes for users. The combinations of these names define unique attributes.

Let's see some examples of basic attributes and their FQAN counterparts:

VO	fred.example.org	/fred.example.org
group	production	/fred.example.org/production
group	replicator	/fred.example.org/replicator
role	VO-Admin	/fred.example.org/Role=VO-Admin
role	Admin	/fred.example.org/Role=Admin
capability	long-job	/fred.example.org/Capability=long-job
capability	large-space	/fred.example.org/Capability=large-space

In a VOMS credential triplets of these basic containers are returned. Since roles and capabilities can not have subcontainers, we order the groups first in an FQAN.

Let's see a subgroup inside replicator:

```
subgroup      optimisation    /fred.example.org/replicator/optimisation
```

We may add a role name to this, which defines the admins of this subgroup, but not the admins of any other group (or attribute):

```
/fred.example.org/replicator/optimisation/Role=Admin
```

In summary a FQAN looks like this:

```
/VO[/group[/subgroup(s)]][/Role=role][/Capability=cap]
```

The name has to match the following regexp:

```
^/<rfc1035>(/[\w-]+)*(/Role=[\w-]+)?(/Capability=[\w_-]+)?$
```

(where `\w` is `[a-zA-Z0-9_]`)

In details:

VO name has to be a well formed DNS name, with the restriction of using only lower case characters[1], i.e. `^([a-z]([a-z0-9-]*[a-z0-9])*\.)*[a-z]{2,4}$` (see also [1])

A simple word, like 'fred' is a valid hostname, so the VO name does not *have* to contain a dot.

group names may contain only word characters and dash.

role names may contain only word characters and dash.

capabilities may contain only word charactes, dash and underscore[2]

[1] the rationale is to be able to use case-insensitive DNS names, but in the case-sensitive database and interface

[2] the rationale of using underscore instead of space is to avoid problems of plain text config files with trailing spaces

2. Representation of Attributes in ACs

Fro RFC 3281 Attribute Certificates we defined a qualifier for *vo-roles* (not the “role” that the RFC 3281 defines), *groups* and *capabilities* in a new attribute, which follows the IetfAttrSyntax:

name:	voms-attribute
OID:	`{ voms 4 }`
syntax:	IetfAttrSyntax
values:	One attribute value only; multiple values within the IetfAttrSyntax

Where `{ voms }` is 1.3.6.1.4.1.8005.100.100[3]

2.1. Example VOMS-AC

A user has these attributes:

```
/fred.example.org
/fred.example.org/production
/fred.example.org/replicator/optimisation
/fred.example.org/Role=VO-Admin
/fred.example.org/production/Role=Admin
/fred.example.org/Capability=long-job
/fred.example.org/Capability=large-space
```

The encoding of these attributes in the attribute certificate is:

```
SEQUENCE {
  OBJECT IDENTIFIER voms-attribute (1 3 6 1 4 1 8005 100 100 4)
  SET {
    SEQUENCE {
      SEQUENCE {
        UTF8String '/fred.example.org'
        UTF8String '/fred.example.org/production'
        UTF8String '/fred.example.org/replicator/optimisation'
        UTF8String '/fred.example.org/Role=VO-Admin'
        UTF8String '/fred.example.org/production/Role=Admin'
        UTF8String '/fred.example.org/Capability=long-job'
        UTF8String '/fred.example.org/Capability=large-space'
      }
    }
  }
}
```

3. Representation of Attributes in XML

All of these attributes are represented in the namespace deisgnated by the following URI: `http://voms.example.org/`

In XACML attributes we don’t distinguish among groups, roles and capabilities at the type/URI level, but only in the content of the attribute: this is a Fully Qualified Attribute Name, as it was described in 1..

[3] The 1.3.6.1.4.1.8005 enterprise subtree is registered for EDG

3.1. Example of VOMS-GACL

The /fred.example.org/replicator/optimisation can read; the /fred.example.org/production/Role=Admin role can read and write as well:

```
<gacl>
  <entry>
     <voms>
        <fqan>/fred.example.org/replicator/optimisation</fqan>
     </voms>
     <allow><read/></allow>
  </entry>
  <entry>
     <voms>
        <fqan>/fred.example.org/production/Role=Admin</fqan>
     </voms>
     <allow><read/><write/></allow>
  </entry>
</gacl>
```

3.2. Example of VOMS-XACML

```
<SubjectMatch MatchId=‘‘urn:oasis:names:tc:xacml:1.0:function:string-equal’’>
  <SubjectAttributeDesignator
    AttributeId=‘‘http://voms.example.org/namespaces/1.0/attribute-id’’
    DataType=‘‘http://www.w3.org/2001/XMLSchema#string’’/>
  <AttributeValue
    DataType=‘‘http://www.w3.org/2001/XMLSchema#string’’>
    /fred.example.org/replicator/optimisation</AttributeValue>
  <AttributeValue
    DataType=‘‘http://www.w3.org/2001/XMLSchema#string’’>
    /fred.example.org/production/Role=Admin</AttributeValue>
</SubjectMatch>
```

A Old Format for Credentials

This section describes the structure of the extensions added by the voms system to the user proxy.

A1. EXTENSION 1

Name:	Voms
Reason:	Return Voms information
OID:	1.3.6.1.4.1.8005.100.100.1
Structure:	
SIGLEN: n	– length of the voms signature in bytes.
SIGNATURE: s	– voms signature
USER: s	– DN of the user’s certificate
UCA: s	– DN of the CA who issued the user’s certificate
SERVER: s	– DN of the server’s certificate
SCA: s	– DN of the CA who issued the server’s certificate
VO: s	– The name of the VO to which the server belongs
TIME1: t	– The start of the validity of this information
TIME1: t	– The end of the validity of this information
DATALEN: n	– The length of the data returned
DATA	– The returned data

A few notes.

1. n means a string representation of a number, s stands for a string, and finally t stands for a ASN1 representation of time.
2. All the values are terminated by a newline character, with the exception of the SIGNATURE: field.
3. The DATA, TIME1 and TIME2 fields do not have the contain only the data, without the name of the field.

Representation of attributes:

If one of the standard queries is made (e.g. not the ’S’ ones) the the data returned is a set of triples with the following syntax:

GROUP: s
ROLE: s
CAP: s

Otherwise, if a ’S’ query is made, the data returned is composed by a set of lines with the following structure:

```
<name of field>: <value of field>
```

In case more than a single Voms server is contacted, there may be multiple copies of the whole structure, starting from the SIGLEN header right to the end of the returned data.

A2. EXTENSION 2

Name:	IncFile
Reason:	Let the user include a specific file into his proxy certificate
OID:	1.3.6.1.4.1.8005.100.100.2
Structure:	A sequence of bytes.

Note that the contents of this field are not the result of a voms request, but do instead contain data specified by the user. The reason for the introduction of this extension was to let a user include important data into its proxy certifiate like, for example, a kerberos ticket.

REFERENCES

[1] Domain Names - Concepts and Facilities, http://www.ietf.org/rfc/rfc1034.txtRFC 1034

[2] Attribute Certificate, http://www.ietf.org/rfc/rfc3281.txtRFC 3281

[3] X509.3 Certificate, http://www.ietf.org/rfc/rfc3280.txtRFC 3280