NetApp™

Go further, faster™

NFSv4 Multi-Domain Access

Andy Adamson
andros@netapp.com
IETF 76 NFSv4 Working Group

Outline

- Motivation
- NFSv4 Authorization Context
- NFSv4 ACL Name
- Name resolution
- LDAP extension
- LDAP Caching Proxy Example
- What's next

Motivation

- The NFSv4 draft addresses all of the pieces of NFSv4 administration each with many options. Stand alone NFSv4 sites choose which options serve their needs.
- Joining NFSv4 servers which can use separate name translation and security services into a multi-domain name space such as the federated file system requires coordination of services.
- NFSv4 deals with two kinds of identities: authentication identities (principals) and authorization identities (users and groups of users). NFSv4 servers must perform two kinds of mapping:
  - Authentication identity <-> Authorization Context
  - On the wire authorization identity <-> On disk authorization identity
- Draft-adamson-nfsv4-multi-domain-access addresses both kinds of mappings describing possible implementation strategies, and specifies a name service for interoperation in a global namespace.

Local Representation of Multiple domains

- Domain: A group of users and computers administered by a single entity, and identified by a DNS domain name.
- Multiple domain access starts at the file server where local ID representation needs to distinguish between local and remote domains.
  - Most installations assign numeric, local identifiers to users and groups using a namespace local to their domain
- A range of suggested solutions for multiple domain representation on disk are presented
  - Large ID: Can express multiple domains on disk using domain-local ID plus a domain ID (Windows SID)
  - Small ID (32-bit POSIX): No room for a domain identifier
- Name resolution (ID <-> name@domain) is required
  - May be less work for Large ID

Multiple Domain and Security Services

- AUTH_NONE can be useful to the multi-domain NFSv4 name space to grant universal access to public data.
- AUTH_SYS uses a host-based authentication model and places the UID and GIDs in the RPC credential and so can only be used in a name space that shares a name translation service.
  - UID/GID collisions occur with multiple name translation services
  - RPCSEC_GSSv3 draft has a modernized replacement for AUTH_SYS
- The NFSv4 mandated RPCSEC_GSS with the Kerberos security mechanism is the only current choice for multi-domain use.
    - X.509-based security mechanisms can also be used. (PKU2U)

NFSv4 Authorization Context

- The NFSv4 server must map the RPCSEC_GSS client principal name (or the GSS security context) to local security information including a domain-local ID, a set of domain-local group IDs and perhaps other user privileges.
- With just using a name service, this means:
  - Contact remote name service over a secure connection to map:
    - RPCSEC_GSS client principal <-> name@domain
    - Obtain a list of group@domain of the group's the user belongs
  - The remote domain name service is the authoritative service for these translations
  - The name@domain and list of group@domain are then mapped to local IDs using the local domain name service or other local means.
- We call this security information an *authorization context* (called an access token in some systems).

NFSv4 Authorization Context

- We define NFSv4 authorization context with the following fields using the GSS-API Naming Extensions name attribute format.
  - draft-ietf-kitten-gssapi-naming-exts
- UserID: principal's global ID and/or local ID mapping, and the name@domain form.
- PrimaryGroupID: global ID and/or local ID mapping for the principal's primary group, and the name@domain form.
- Groups: an array of group IDs for the groups that the user is a member of, in global ID and/or local ID form, and in name@domain form
- YTD field(s)
  - privileges and authorizations granted to the principal
  - Multi-level security label range/set
  - Implementation specific items

NFSv4 Authorization Context Determination

- The NFSv4 access token SHOULD be obtained via the per GSS-API mechanism naming extension named attribute interface.
  - There is an MIT Kerberos implementation under development.
- If the GSS-API attribute interface is not available:
  - Obtain the access token information from the authenticating credentials of the principal
    - Kerberos PAC
  - Map the RPCSEC_GSS client principal to a local user account, and then lookup that user's account access token information from the user's domain name services.
    - See LDAP Extension Example

NFSv4 ACL Name

- NFSv4 owner, owner_group, acl, dacl and sacl attributes represent a file object's authorization metadata. The NFSv4 owner, owner_group and ACE 'who' field we call the NFSv4 ACL name.
- The on-the-wire name@domain form of the NFSv4 ACL name for users and groups gives a level of indirection that allows a client and server to translate their local ID representation to a common syntax.
- Multi-domain capable sites need to meet these requirements in order to ensure clients and servers can map name@domain to internal representations reliably:
  - name@domain MUST be unique within the DNS domain
  - Every local representation of a user and a group MUST have a name@domain, and it MUST be possible to return the name@domain for any identity stored on disk, at least when required infrastructure such as name services are on line.

Multiple Domain Name Resolution

- A domain's name service is authoritative for:
  - Join/Leave/Rename (validity of name@domain)
  - Authorization Context Information mappings
- Multiple domain capable sites therefore need to do name service lookups in various domains
  - Remote services may not always be available
- Site administrators may wish to maintain local caches of key attributes (e.g. a caching proxy).
  - This is recommended
- Domains in a federated namespace may provide each other with LDAP LDIF delta feeds to maintain cached LDAP contents up to date.

Multiple Domain Name Resolution

- To support multiple domain name resolution, implementations are REQUIRED to support the use of LDAP with the RFC2307 schema as a name service.
  - To support authorization context information lookup
  - Other schemas are allowed
- Each Domain (local and remote) has a corresponding base DN as follows
  - Strip the trailing dot (.), replace all dots with “,DC=“ , prepend “DC=“to the resulting string
  - foo.bar.example.com becomes DC=foo,DC=bar,DC=example,DC=com
- This convention is REQUIRED. Other conventions allowed if domainname<->base DN mapping is published

LDAP Extension

The gSSAuthName attribute provides a translation between the domain-local ID and (multiple) GSS security principals.

attributetype (1.3.6.1.4.1.250.10.6

NAME (‘gSSAuthName’)

DESC ‘GSS-API principal name exported token’

EQUALITY bitStringMatch

SYNTAX 1.3.6.1.4.1.1466.115.121.1.6)

LDAP Extension

The gSSPrincipal objectclass allows for the gSSAuthName attribute to be associated with a posixAccount.

attributetype (1.3.6.1.4.1.250.10.7

NAME (‘gSSPrincipal’)

DESC ‘GSS Principal Name’

SUP posixAccount

MAY(gSSAuthName))

LDAP Caching Proxy Example

- Here is the local domain (sample.com) LDAP name service caching the remote domain (university.edu) rfc2307 posixAccount information with the gSSAuthName attribute.

dc=com, dc=sample, ou=people

<All rfc2307 people entries for sample.com>

uid=bob, uidNumber=2501,

gSSAuthName=bob@SAMPLE.COM

dc=edu, dc=university, ou=people

<All rfc2307 people entries for university.edu>

uid=alice, uidNumber=3888,

gSSAuthName=alice@UNIVERSITY.EDU

- The cached university.edu information stored in sample.com's LDAP name service needs to be validated on a regular basis.
  - Perhaps with an LDIF feed from university.edu

What's Next

- Drill into NFSv4 Authorization Context definition
- Complete LDAP extensions
  - ID mapping (remoteID <-> localID)
- Additional text on remote groups

Questions?