NFSv4 Multi-Domain Access

Andy Adamson
andros@netapp.com
Connectathon 2010

Outline

- Motivation
- Security and NFSv4 Authorization Context
- Local ID Representation and name resolution
- LDAP example
- What's next

Motivation

- Many administrative options in the NFSv4.0/4.1 draft.
  - Stand alone NFSv4 sites choose which options serve their needs.
- Not all choices will work across a multi-domain namespace
  - Mainly due to separate name translation services
- 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 name service configurations for interoperation in a multi-domain namespace.
  - Co-authors Kevin Coffman and Nico Williams

NFSv4 Domain for Multi Domain Access

- NFSv4 Domain: A group of users and computers administered by a single entity, and identified to NFSv4 by a DNS domain name.
  - Multiple DNS domains
  - Multiple security services
  - Single name translation service
- Multi-domain capable sites need to translate 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
  - It MUST be possible to return the name@domain for any identity stored on disk

Multi Domain Security Services

- AUTH_NONE can be used - access to public data.
- AUTH_SYS can only be used in a file name space that shares a name translation service.
  - Places the UID and GIDs in the RPC credential
  - UID/GID collisions occur with multiple name translation services
  - RPCSEC_GSSv3 draft has a modernized replacement for AUTH_SYS which could be used
- The NFSv4 mandated RPCSEC_GSS with the Kerberos security mechanism is the only current choice for multi-domain use.
  - X.509-based security mechanisms could also be used. (PKU2U)
- Cross realm trust between NFSv4 Domains is required
  - Except for unmapped 'nobody' access

Cross Realm Trust

- Kerberos cross-realm trust means that any authenticated user can obtain service tickets in the foreign realm
  - Turns on authentication to all Kerberized services
  - Requires that all Kerberized services provide access control
- X.509 cross realm trust is per service
- Each X.509 service in the foreign realm needs a self-signed CA certificate
  - Certificate per NFSv4 server
- In all cases, NFSv4 access is controlled via ID mapping and ACLs
  - No ID mapping -> no (or limited) NFSv4 access

Authorization Context

- The NFSv4 server must map the RPCSEC_GSS client principal name (or the GSS security context) to local security information
  - A domain-local ID
  - Set of domain-local Group IDs
  - Other privileges
- We call this security information an ***authorization context*** (called an access token in some systems).
  - Remote domain the authoritative source
- We define an NFSv4 Authorization Context using the GSS-API Naming Extensions name attribute format – the NFSv4 version of the Windows Kerberos PAC
  - Uses name@domain instead of SIDs
  - See draft-ietf-kitten-gssapi-naming-ext

NFSv4 Authorization Context

- ***UserID***: principal's global ID and/or user domain ID mapping, and the name@domain form.
- ***PrimaryGroupID***: global ID and/or user domain 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 user domain 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

- The NFSv4 authorization context information SHOULD be obtained via the per GSS-API mechanism naming extension named attribute interface.
  - Still need to translate the name@domain into local domain IDs
  - There is an MIT Kerberos implementation under development.
- Else use the Kerberos PAC if available
  - May need to translate the KPAC SIDs into local doman IDs
- With just using a name service
  - The remote domain name service is the authoritative service for these translations
  - Contact remote name service over a secure connection
  - Map a principal@REALM to name@domain (see later slides)
  - The name@domain and list of group@domain are then mapped to local IDs using the local domain name service or other local means.

Multi Domain Name Resolution

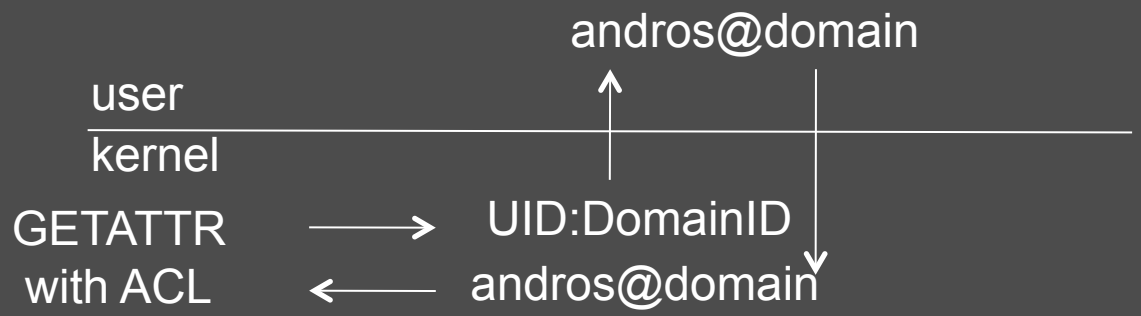
- A domain's name service is authoritative for:
  - Join/Leave/Rename (validity of name@domain)
  - Authorization Context Information mappings
- Multiple domain capable sites therefore need to do name service lookups in various domains
  - Remote services may not always be available
- Site administrators 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.

Local ID Representation

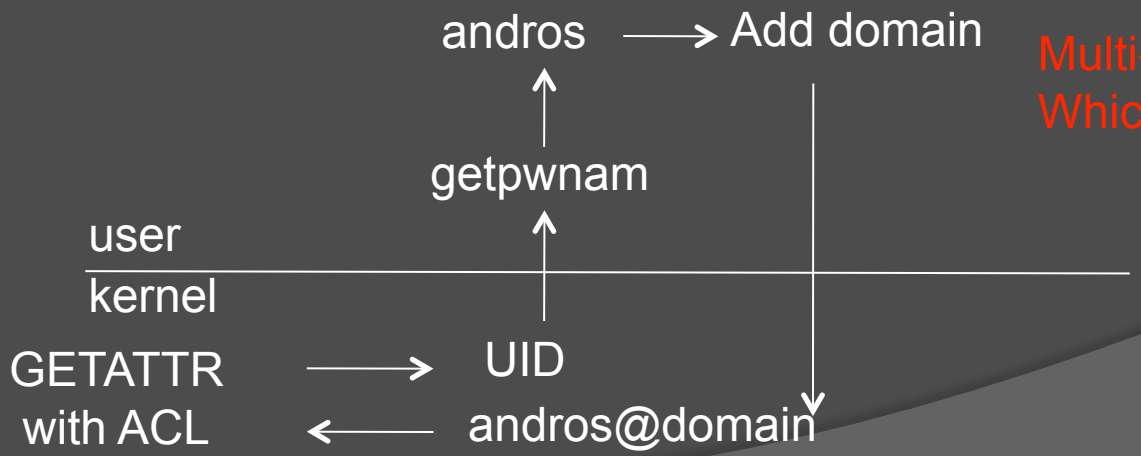
- Multiple domain access starts at the file server where local ID representation needs to distinguish between local and remote domains.
  - Most installations assign numeric, local identifiers to users and groups using a namespace local to their domain
- A range of suggested solutions for multiple domain representation on disk are presented in the draft.
  - 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

Local ID Representation

Large ID



Small ID

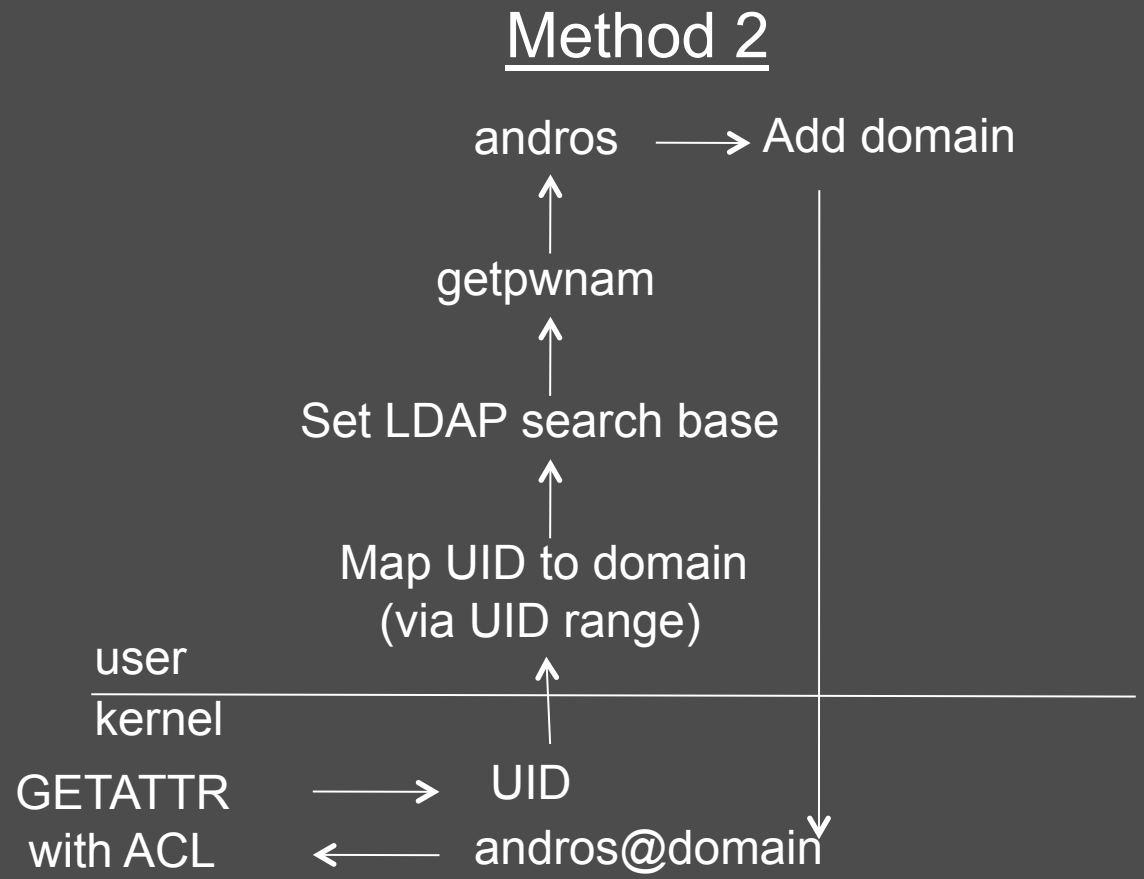


Multi-domain:
Which domain to add?

Small ID Domain Mapping

- Identified two methods of translating a small UID into a name@domain
- **Method 1**: CITIs umich_ldap schema NFSv4Name attribute which is associated with the uidNumber and holds the name@domain
  - Distributed in fedora
  - Requires new ldap seach, can not use NSS getpwXXX functions
- **Method 2**: Reserve a UID number range and add an LDAP hierarchy per remote domain.
    - Determine domain via range
    - Change LDAP search base
    - Use NSS getpwXXX functions
    - Preferred method

Small ID Domain Translation



Multi 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

What About ls -l ?

- Local domain UIDs are translated to a user name
  - Often by getpwnam
- Remote users have UID, but the '@' character is not permitted in many implementations.
  - Remote UIDs are not translated.
- Suggest a substitute for the '@' character in name@domain
  - Such as '-' ??
- Assign remote users a username (PosixAccount uid) of the form 'name<substitue char>domain'
  - e.g name-domain

Multi Domain Kerberos Principal Translation

- A common convention is to name a Kerberos Realm as the @REALM is the upper case of the DNS domain
- If this convention is followed, and the DNS domain is used as the NFS4 domain, then the Kerberos principal <-> UID translation is direct.
- If this convention is not followed, or if there are multiple security realms in an NFSv4 domain, an additional LDAP attribute needs to be associated with the UID

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 Example

- Here is the local domain (sample.com) LDAP name service representing 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

- Want to be added as an IETF NFSv4 working group item
- Drill into NFSv4 Authorization Context definition
- Address remote groups
- Complete LDAP extensions

Questions?