NFSv4 Multi-Domain Access

Andy Adamson

andros@netapp.com

IETF 77 NFSv4 WG Meeting

March 23, 2010

History

- ***Draft-adamson-nfsv4-multi-domain-access-01*** was presented at IETF 75.
  - Co-authored by Kevin Coffman
  - Based mainly on experience at the University of Michigan
  - NFSv4 multi-domain work at CITI for the Tri-Labs
- ***Draft-adamson-nfsv4-multi-domain-access-02*** was presented at IETF 76.
  - Nicolas Williams joined as a co-author
  - Added his extensive corporate multi-domain experience

Outline

- The problem
- In a perfect world
- Where we are today
- What’s next

Authentication Identity Translation

- NFS servers need to do an authentication identity translation to determine file access rights

Authentication identity -> Local representation of identity and associated authorization Information

- Authorization information includes the authenticated identity's primary group and a list of groups the identity is a member of.

Authentication Identity Translation

- AUTH_SYS rpc credential contains all the information required by the server to make authorization decisions
  - UID
  - Primary GID
  - List of other GIDs
- The AUTH_SYS rpc credential embodies the translation, no more work to be done.
  - Except a common UID/GID to user/group name mapping across NFSv4 clients and servers ***which forbids it's use in a multi-domain namespace.***

The Problem

- Although AUTH_GSS security mechanisms are designed to work in a multi domain environment, the ***security services don't include authorization***
- Inconvenient at best for local domain access
- Really bad for multi-domain access

The Problem

- As a result, the NFSv4 server only gets the authentication identity at GSS context creation, which needs to be mapped to the file system's identity representation
- The primary group and list of groups must be obtained in a separate step outside of the RPCSEC_GSS user authentication to the server and need to be mapped…
- For local domain users, this problem is mostly solved as the same information is required for local file system access.

The Problem

- Adding multi-domains exasperates the problem
- A multi domain NFSv4 server must obtain authorization information from an authoritative service in a non-local domain
- The authorization information needs to be in the global NFSv4 name@domain format so that it can be mapped from a remote to a local representation
- Identity and group representation in exported file systems needs to be domain-aware

In The Perfect World

- NFSv4 authorization information would be embedded in per GSS security mechanism contexts by the security authority creating the context payload
- Authorization information for local domain access would be in a form that the NFSv4 server could use without any translations
- Authorization information for remote domain access would be in the NFSv4 name@domain format

In The Perfect World

- There would be a GSS security mechanism independent GSS interface to access the authorization information
- All exported file systems local ID representations would be ‘domain aware’

Where We Are Now

- Most NFS sites use AUTH_SYS, and will continue to do so as they move to NFSv4
- Most NFS sites that do use Kerberos use a single REALM where the REALM == DNS name
  - The principal@REALM is interpreted as a local domain username (username@DNSdomain)
  - The local domain username (because it is equivalent to the Kerberos principal) is used to lookup authorization information (just like UNIX login).

What’s next

- The first use of the NFSv4 (Federated) name space will probably be within a single administrative domain with no extra authorization information mapping required

- The first *multi domain* NFSv4 (Federated) name space will join two sites that already use Kerberos and are already exporting file systems with domain-aware identity and group representation.

What’s next

draft-adamson-nfsv4-multi-domain-access

- Describe the smallest change to allow exiting NFS Kerberos enabled sites to become NFSv4 multi-domain capable.
- Describe multi-domain local representation solutions for file systems
- Describe methods that NFSv4 servers can use to obtain remote user authorization information for GSS security mechanisms when GSS authorization APIs and/or authorization information in the GSS context are not present

What’s next

draft-adamson-nfsv4-multi-domain-access

- Propose content for a per GSS security mechanism NFSv4 GSS Authorization Context extension
  - Authorization information in a form suitable for a NFSv4 server to use for local and remote user access
- Use the *draft-ietf-kitten-gss-naming-ext* interface to access the per GSS security mechanism authorization extensions

Questions?

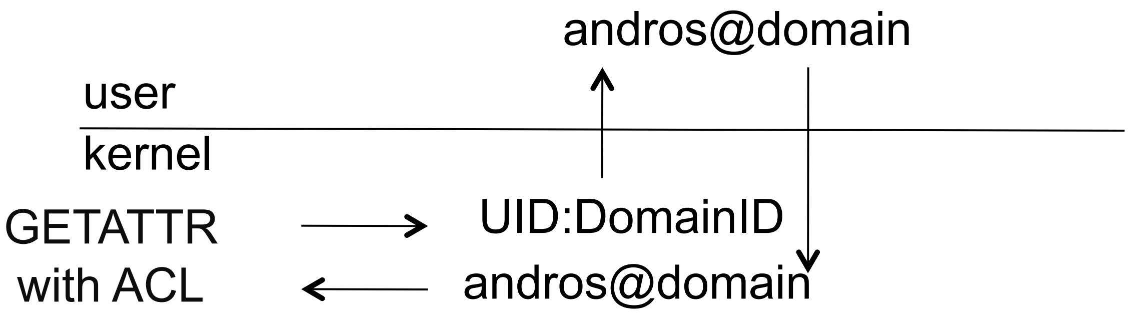
Background

Local ID Representation

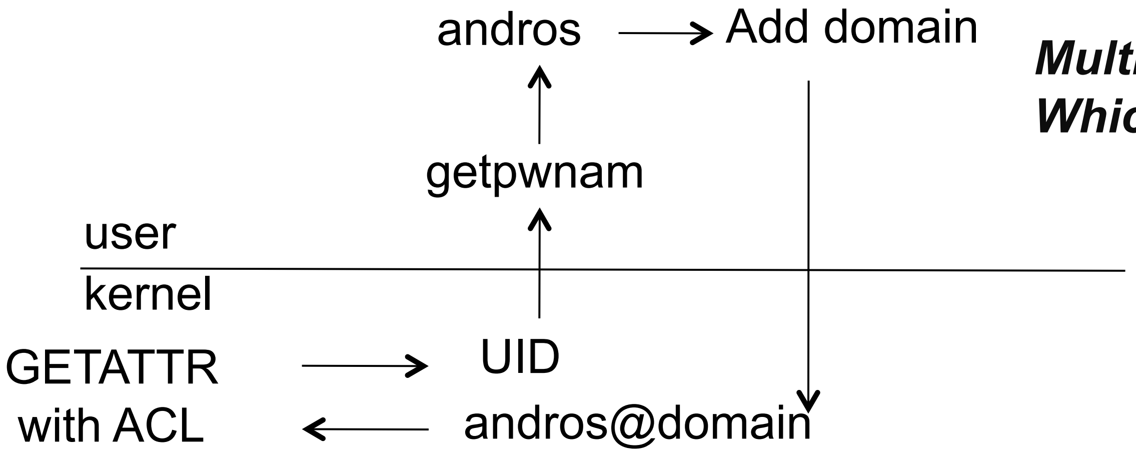
- Most installations assign numeric identifiers to users and groups using a namespace local to their domain
- A range of suggested solutions for multiple domain representation on disk are presented in the draft.
  - Large ID: Can express multiple domains on disk using domain-local ID plus a domain ID (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

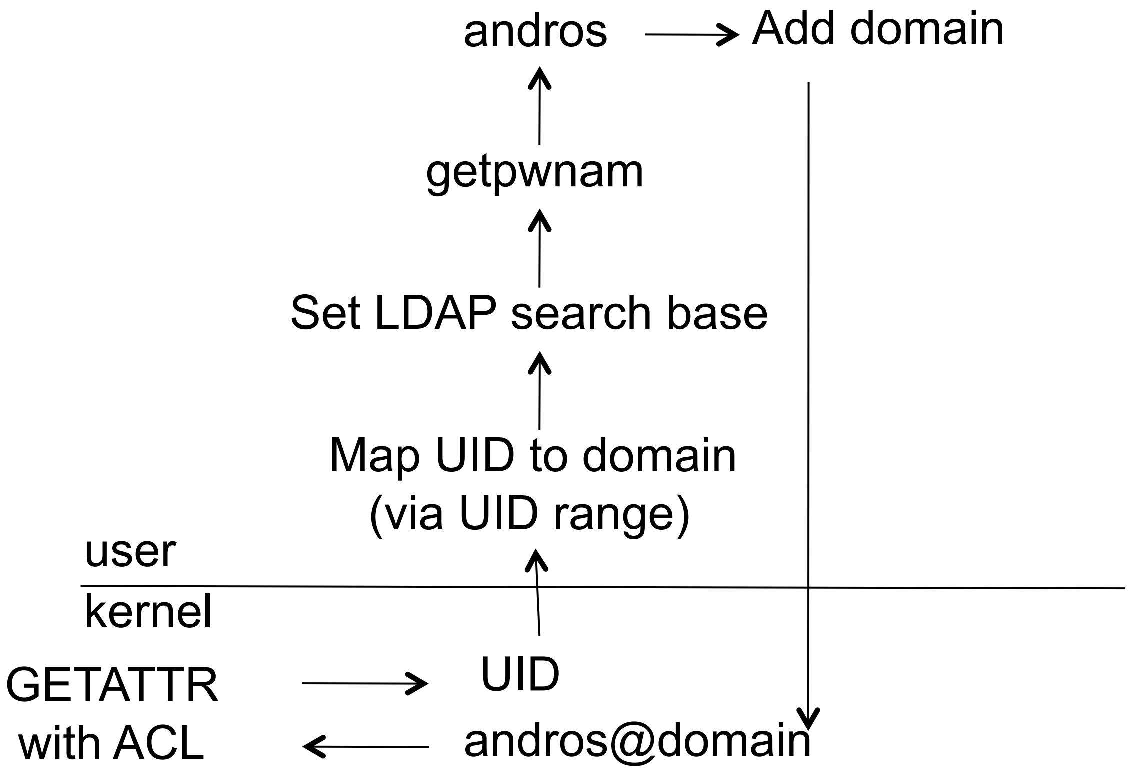
- **Method 1**: Translating a small UID into a name@domain
- University of Michigan CITI umich_ldap schema NFSv4Name attribute which is associated with the uidNumber and holds the name@domain
  - Distributed in fedora
  - Requires new ldap seach, can not use NSS getpwXXX functions

Small ID Domain Mapping

- **Method 2**:Translating a small UID into a name@domain
- 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

Method 2



andros

Add domain

getpwnam

Set LDAP search base

Map UID to domain (via UID range)

user

kernel

GETATTR with ACL

UID

andros@domain

NFSv4 Domain

- NFSv4 Domain is the building block of multi domain namespaces and is defined as follows:

A group of users and computers administered by a single entity, and identified to NFSv4 by a DNS domain name.

  - Can include multiple DNS domains
  - Can include multiple security services

Multi-domain name@domain Rules

- 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

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

Cross Realm Trust

- 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

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

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) )
```