NetApp

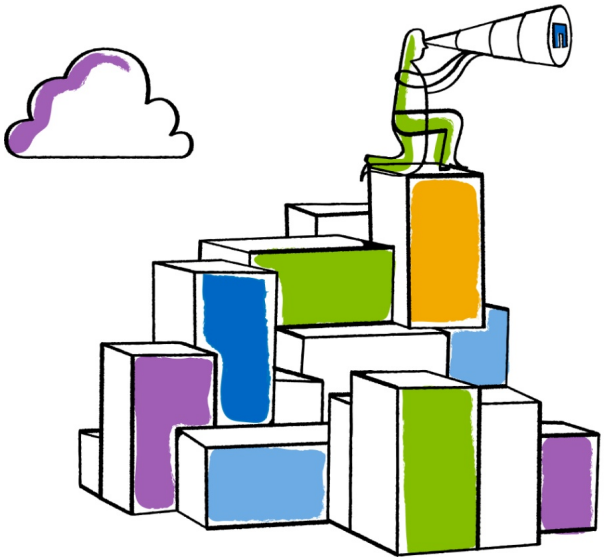
Go further, faster®

NFSv4 Multi-Domain FedFS Requirements

William A. (Andy) Adamson

andros@netapp.com

IETF 88 Vancouver



Motivation

- FedFS is done
- Targets FedFS use of NFSv4 – Defines the “NFSv4 multi-domain federated file-system”
- NFSv4 protocols are defined in a manner that allows administrators a lot of configuration freedom
  - NFSv4 domain where each client and server has it’s own local idea of name ⇔ ID mapping (/etc/passwd, /etc/group)
- Some of these allowed configurations will not work in a multi-domain environment

Motivation

- While the requirements in this draft may be 'obvious' they still need to be said somewhere
- Can not join 2 NFSv4 Domains under FedFS without following these requirements
- The requirements center around the issues of mapping between RPCSEC_GSS security principal names or NFSv4 name@domain and Local ID representations
  - Must avoid collisions in a multi-domain environment

Name Service

- Provides the mapping between {NFSv4 domain, group or user name} and {NFSv4 domain, local ID} via lookups used by NFSv4 servers and clients
  - name@domain ⇔ Local ID in local NFSv4 domain
  - principal@REALM ⇔ Local ID in local NFSv4 domain
- Can be applied to local or remote domains or Kerberos REALMs
- Often provided by a Directory Service such as LDAP

Multi-domain Capable File System

- File system with an ID form that can represent identities from local and remote domains
  - E.G can interpret a domain component
- SSID based file systems are an example
  - Usually not exported by NFS ☺
- 32 bit POSIX based file systems are *not* an example
  - Vast majority of exported NFS file systems
  - Strip off NFSv4 domain portion of name@domain and REALM portion of principal@REALM to map user-principal to a UID
  - Methods to enable POSIX multi-domain out of scope of this draft

Multi-domain capable NFSv4 domain

- A set of users, groups and computers running NFSv4 protocols employing a single name service, and identified by a unique NFSv4 domain name
- All servers in a multi-domain capable NFSv4 domain export multi-domain capable file systems

NFSv4 multi-domain federated file-system

- Uses FedFS to join multiple NFSv4 domains
- Each NFSv4 Domain is multi-domain capable
  - All NFSv4 servers export multi-domain capable file systems

Name@domain Constraints

- Domain portion of name@domain MUST be unique within the FedFS NFSv4 multi-domain namespace
- The name portion of name@domain MUST be unique within the specified NFSv4 domain
- Every local representation of a user and of a group MUST have a canonical name@domain
- It must be possible to return the canonical name@domain for any identity stored on disk
  - Caveat name services are on-line

Multi-Domain RPC Security Constraints

- The RPC security flavor MUST have a domain (or realm) component in it's security identities
  - Required to avoid cross domain collisions
  - AUTH_SYS: No domain component, so can not be used
- Security flavor is REQUIRED to employ a method of cross domain trust
  - Required to enable recognition of remote principals
  - RPCSEC_GSS with Kerberos or PKI (PKU2U) are examples
- No Credentials (AUTH_NONE) is the exception

Resolving Cross Domain Authorization

- After confirming the identity of an RPC principal, the NFSv4 server needs to obtain, in a secure manner, the authorization information of the RPC principal from an ***authoritative source*** to determine file access capabilities
  - username, userUID, group membership, etc
  - Just like the local domain case
- Define what is ‘authoritative’ for remote domain principals
  - The remote domain’s name service is one example

Resolving Cross Domain Authorization

- draft-adamson-nfsv4-multi-domain-federated-fs-reqs-03 goes on to describe the three ways remote principal authorization information can be obtained
  - Mechanism specific GSS-API authorization payload
  - Local name-service is authoritative for remote principal due to security agreements and regular update feeds from remote site
  - Direct query to remote site name service
- Probably should leave these details to a best-practices draft

Issues

- Is this draft subject useful? I think so….
- Distill down to requirements only
  - Reduce ‘Resolving cross domain authorization’ to a requirement, perhaps remove section
- A lot of over-used terminology that needs to be clarified for use in the draft – e.g. ‘domain’
- What about multi-domain groups?
  - Any requirements here?
- Help by reviewing!!