DCache and it's security models development

Owen Synge
DCache Team member

Structure Of This Talk

- Introduction
- Authentication
- Authorization
- Accounting
- Auditing
- Summary

Introduction

- Dcache
  - Mass storage system cache management
    - Provides a site based storage cluster
    - Manges cluster
  - Written primarily HEP community
    - Low security requirements
    - Performance a priority
    - Long production history
  - Expanding to suit disk only services
    - New service Resilience Manager.

Authentication

- Dcache doors (Note: all are optional)
  - Certificate based Authenticated.
    - xrootd (POSIX like file access)
    - nfs v4 native support (POSIX like file access)
    - SRM (data management API V1 and soon V2)
    - GSIFTP (WAN data transfer API)
    - gsidcap (POSIX like file access)
  - RSH style Authentication
    - dcap (POSIX like file access)
    - xrootd (POSIX like file access)

Authentication RSH

- Why do we have rsh authentication still?
  - Usefull as it maps to UNIX 1:1
  - Some uses want high performance
  - Which doors
    - dcap,xrootd
- What can I do to make them secure?
  - Switch them off.
  - Set them read only.
    - for dcap and xrootd

Authentication GSI

- Certificate based access
  - Most doors : SRM, GSIFTP, GSIDCAP
- Uses GridMap files.
  - All users and “VO” mapped simply
- Not scalable.
  - As “VO” number grows users may be in multiple VO's. Mapping must be downloaded
- DCache's Implementation is poor
  - CRL's not honored

Authentication SAML/VOMS

- Improvement over GSI
  - VO/Group mappings provided before cert reaches DCache.
  - Supported by SRM and GSIFTP
  - Allows users to be in multiple VO's
  - Removes need for dynamic list management
- DCache Implementations
  - Maps to UID,GID based upon VO and Group

Authentication in gPlazma Cell

- If authorization fails or is denied, attempts next method

dcachesrm-gplazma.policy:

```
# Switches"
saml-vo-mapping="ON"
kpwd="ON"
grid-mapfile="OFF"
gplazmalite-vorole-mapping="OFF"

# Priorities
saml-vo-mapping-priority="1"
kpwd-priority="3"
grid-mapfile-priority="4"
gplazmalite-vorole-mapping-priority="2"
```

Authentication Future

- Support for SAZ
  - Acts as a client to Site AuthZ server
  - When
    - Release 1.8.0
- Dynamic UID,GID output
  - Currently DCache UID matches host UID
  - Virtual UID/GID allows for richer Authorisation
  - When
    - Release 1.7.1

Authorization and Name services

- Dcache couples name service and authorization rules datastore
- The current implementation (PNFS)
  - Unix UID/GID based
  - Requires host UID/GID available
  - Not scalable to Grid world
    - Files can have only one GID
    - No way to support privileges within a VO
      - production/user distinction

Authorization and Chimera

- Chimera Name space service
  - Supports ACL's (POSIX Style)
  - Uses Virtual UID/GID model
    - Mapped 1:1 from unique DN/FQN or UID GID
    - All resources have vUID/vGID
  - Action is extended (Not just RWX as in UNIX)
  - Built in hierarchy support.
    - All directories in a tree tested.
  - NFSv4 model

Authorization and Chimera 2

- Database based
  - UID/GID will not be needed on host
  - Each resource will have single vUID
  - Each resource will have multiple vGID
    - But a primary vGID to enhance UNIX interoperability
- Interoperability with NFSv4 clients has been shown for development versions.
  - for windows, sun and linux

Accounting

- Accounting will remain at the VO level for the next year.
  - VO provide resources on the pool level
  - VO are given to storage at the pool level
- Two types of Accounts expected by LHC
  - Production users
  - Normal Users
- Quota implementation not seen as an immediate priority for HEP.

Auditing before last year

- Dcache is a modular service
  - Each module provided separate logging.
  - Admins must write parsers to intelligently gather data from many log files
  - Was a challenge to find who wrote what file
  - Was not acceptable on one site.
    - Mass deployment made potential dangers worse

Auditing

- Dcache Logs to a RDBM
  - audit queries have been developed.
    - All files by user ID
  - Most queries are by write requests
    - Who wrote what file
    - Who wrote how much data in what time frame
- Legal requirement for global deployment
  - Must provide more information

Summary

- Dcache is progressing towards Grid security in a use case driven way.
  - This is not a small change from a UNIX model
  - Authentication is still evolving.
  - ACL support is a requirement
  - Auditing is a requirement for global deployment
- DCache is soon to be have the minimum of security functions to be a valid Grid Storage system