Kerberized Lustre 2.0
over the WAN

Josephine Palencia, Robert Budden, Kevin Sullivan
Pittsburgh Supercomputing Center
Content

- Kerberos V5 primer
  - Authenticated lustre components
- Setup: Single Kerberos realms->Cross-realm
  - Partners: SDSC@TeraGrid, NRL, Uflorida
- Distributed OST/OST pools
  - Arranging user directories
- Lustre 2.0 tests
  - Available systems: Quota, ACL
  - Kerberized scp/kftp/gridftp with konFUSEd
- Current constraints
  - Integrating with kerberized NFS4
- Summary
- References
## Kerberos V5 Primer

**Client**
- authenticates itself to KAS,
- demonstrates to TGS that it's authorized to receive a ticket for a service
- demonstrates to SS that it has been approved to receive service

**Terms**
- U = user, CL = client
- KAS = kerberos authentication server
- TGS = ticket granting server
- SS = service server

---

© 2010 Pittsburgh Supercomputing Center
### Authenticated lustre components

#### Shift among kerberos flavors during mounts
- Preliminary IO benchmark comparing different flavors to look at performance overhead by kerberos

<table>
<thead>
<tr>
<th>Flavor</th>
<th>Auth</th>
<th>RPC message</th>
<th>Bulk Data Prot protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td></td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>plain</td>
<td></td>
<td>null</td>
<td>checksum(adler32)</td>
</tr>
<tr>
<td>krb5n</td>
<td>GSS/krb5</td>
<td>null</td>
<td>checksum(adler32)</td>
</tr>
<tr>
<td>krb5a</td>
<td>GSS/krb5</td>
<td>partly integrity</td>
<td>checksum(adler32)</td>
</tr>
<tr>
<td>krb5i</td>
<td>GSS/krb5</td>
<td>integrity</td>
<td>integrity(sha1)</td>
</tr>
<tr>
<td>krb5p</td>
<td>GSS/krb5</td>
<td>privacy</td>
<td>privacy(sha1/aes128)</td>
</tr>
</tbody>
</table>

lustre_root/goldeneye.jwan.teragrid.org@TERAGRID.ORG (AES-256 CTS mode with 96-bit SHA-1 HMAC)

nfs/goldeneye.jwan.teragrid.org@TERAGRID.ORG (AES-256 CTS mode with 96-bit SHA-1 HMAC)

(lustre_mgs, lustre_mds, lustre_oss, lustre_root)

---

© 2010 Pittsburgh Supercomputing Center
Setup

- Lustre 2.0 Alpha 5 (CentOS5)
- Release: 1.9.280
- Kernel: 2.6.18-128.7.1
- Kerberos auth: krb5p
Cross-Realm kerberos authentication work
NSF funding to set up kerberized Lustre 2.0 for OSG

Secure Lustre 2.0 filesystem will be based at University of Florida

Project involves setting up a kerberos infrastructure and choosing a kerberos realm.
Distributed OST/(optional) OST pool

- decentralize/distribute the secure Lustre WAN storage across several organizations/different sites
- sites write to local OST pools but the filesystem is visible on the WAN
- data striped
  - across PSC dist OSTs/pool
  - across SDSC dist OSTs/pool
  - across both sites OSTs
Distributed OST/OST pool

Best way to arrange TeraGrid user directories in a lustre WAN with distributed OST/OST pool?

1. Sysadmins: OSS-centric directory arrangement
   Data is 'striped' to local OSS contributed by each site
   
   so /jwan/$SITE/users
   
   For the 6 sites and user joe (N-> 0-9),
   /jwan/psc/N/joe
   /jwan/sdsc
   /jwan/taccs
   /jwan/iu
   /jwan/nics
   /jwan/ncsa...

2. Users: user-centric way of seeing their data in specific machines
   
   so /jwan/$USER/$SITE/$MACHINE
   
   /jwan/joe/psc
   /psc/pople
   /jwan/joe/tacc
   /tacc/ranger
   /tacc/lonestar
   /jwan/joe/nics
   /nics/kraken...
Distributed OST/OST pool

Best way to arrange TeraGrid user directories in a lustre WAN with distributed OST pool?

3. Combining both and adopting something similar to AFS

so /jwan/users/$USER/$PSC/$machine symlink to /jwan/$SITE/N/$USER/$MACHINE

More specifically,

/jwan/users/joe/psc/pople --> /jwan/psc/users/N/joe/pople

where

/jwan/users/$USER and
/jwan/users/$USER/$SITE level directories are both read-only and highly-available to users
Constraints: Part II

- Lustre 2.0 – only interoperable with (non-kerberized) Lustre 1.8
  - Mount Lustre 1.8.2 on a kerberized Lustre 2.0 client without disabling kerberos on the Lustre 2.0 servers

- Native clients, OSTs- must be running lustre 2.0
  - same release (kernel/lustre rpms- available)

- Other sites/partners don't have (will not have) the kerberos infrastructure

How to make kerberized lustre 2.0 accessible and usable in the current framework?
Kerberized scp/kftp/gridftp: konFUSEd

pople.psc.edu

login Node

File System

/boot
/usr
/home
/tmp

file system

kinit joe@TERAGRID.ORG

krb5cc_1000

written to /tmp

“konfused” (fuse interceptor)

ticket propagated to /tmp

thunderball.jwan.teragrid.org

Lustre WAN Client

• GridFTP
• HPN SSH Server
• KFTP Server

File System

/boot
/lwan
/tmp
Kerberized scp/kftp/gridftp: konFUSEd

[joeuser@tg-login3.pople.psc.teragrid.org:~]$ scp file thunderball.jwan.teragrid.org:/jwan/users/joeuser/test

joeuser@thunderball.jwan.teragrid.org's password:
scp: /jwan/users/joeuser/test: Operation not permitted

[joeuser@tg-login3.pople.psc.teragrid.org:~]$ module load konfuse

[joeuser@tg-login3.pople.psc.teragrid.org:~]$ env | grep KRB
KRB5CCNAME=/konfusefs/krb5cc_20033

[joeuser@tg-login3.pople.psc.teragrid.org:~]$ kinit joeuser@TERAGRID.ORG
joeuser@TERAGRID.ORG's Password:

[joeuser@tg-login3.pople.psc.teragrid.org:~]$ scp file thunderball.jwan.teragrid.org:/jwan/users/joeuser/test
file 0% 0 0.0KB/s 0.0KB/s ---:--- ETA file 100% 611KB 611.0KB/s 611.0KB/s 00:00
Max throughput: 611.0KB/s

[joeuser@tg-login3.pople.psc.teragrid.org:~]$ exit
Available resources @ PSC

- 11,115 Teragrid Accounts auto-synced from TGCDB (Teragrid Central Database)
- Any Teragrid user can login to VM kerberized lustre 2.0 client goldeneye.jwan.teragrid.org and access the filesystem
- run tests: Lustre 2.0 quota, ACLs
- request writes to local PSC OSTs (pool), remote SDSC osts (pool), combination of local and remote OSTs (pool)
- Perform data transfer on Lustre 2.0 fs
  - Gridftp/Gsiscp/Kftp with another VM data target (databox) thunderball.jwan.teragrid.org
Contribute resources

Active participants:

Index of ftp://ftp.psc.edu/pub/jwan/Lustre-2.0-alpha/

- Kernel/Lustre rpms
- Kerberos infrastructure (with/without)
  - Sites already in the TeraGrid
  - Sites not in TeraGrid (untested)
Kerberized NFS4

<table>
<thead>
<tr>
<th>Lustre2.0</th>
<th>NFS4</th>
<th>User Access to /lwan-nfs4</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>sec, krb5 (i, p)</td>
<td>✔</td>
</tr>
<tr>
<td>krb5 (n, a, i, p)</td>
<td>-</td>
<td>✗ /lwan-nfs4 remains mounted but inaccessible to users</td>
</tr>
</tbody>
</table>

Exports /lwan2.0 via NFS4 kerberos - enabled

mount /lwan2.0 via Lustre with auth krb5p
lwan.psc.flavor.default = krb5p

mount /lwan2.0-nfs4 via NFS4 (sec = krb5p)
Kerberized NFS4

- Kerberized Lustre 2.0
  - keyring
  - pipefs (remnant)

- Kerberized NFS4
  - pipefs

- Keyring (key ctrl)
  - Lustre
  - + mechanism for key forwarding
  - NFS4
    - keys?

Lustre 2.0
- lwan.srpc.flavor.default =

NFS4
- sec =
  - sec
  - krb5
  - krb5i
  - krb5p

- Keyring
  - Keys
    - Tgt
    - lustre_mds
    - nfs
  - null/plain
    - krbn
    - krb5a
    - krb5i
    - krb5p
Constraint: Part II

- Lustre kerberos in Single Kerberos Realm
  - necessary to sync all accounts on MDS and all clients
  - user-land: 3-fold check (uid, gid, kerberos principal)
  - lustre ID mapping turned off for single kerberos realm

- Lustre kerberos- Cross-realm
  - lustre ID mapping enabled
Summary

- Workable secure Lustre 2.0 system with constraints
- Working on kerberized NFS4 and Lustre 2.0
- Lustre kerberos realm: Single->Cross realm
  - Both user-land and systems
  - Key management
References

TeraGrid Kerberized Lustre 2.0 Wiki

References
PSC Kerblustre Wiki
https://wiki.psc.edu/twiki/view/KerbLustre/WebHome