High Level Design of Cient-OSS Connection

Peter Braam, Eric Mei

Mar 2, 2005

1 Requirements

- Establish gss connections between clients and OSS.
- Establish gss connections between servers.

2 Functional Specification

In Lustre system, there are several kinds of connections and security options can be chosen separately:

- between client and MDS's
- between client and OSS's
- between MDS's
- between MDS's and OSS's

Currently we are able to establish secure connections between the client and MDS's, simply by adding a mount parameter 'sec=sec_flavor', here sec_flavor could be "null", "krb5i" or "krb5p" for this moment. Now we also need the secure connections between client and OSS's also be an option, to prepare for the coming object security features. So the original mount option 'sec' will be break into 2 options: 'mds_sec' and 'oss_sec'.

- mount.lustre should be able to recognize options 'mds_sec=sec_flavor' and 'oss_sec=sec_flavor'.
- lmt should be able to add 'mds_sec' and 'oss_sec' into xml file and recognizable by lconf. And lconf should be able to write this info into config log with option –write-conf.

Usually we consider MDS and OSS are trusted nodes, but networks are normally not secure. So connections of MDS <=> MDS and MDS <=> OSS must be secure in most cases. We should also provide security on connections between servers.

For inter MDS's and MDS's to OSS's, We provide options for lconf and lmt, just like client <=> OSS's case:

- lconf should be able to recognize options '–inter_mds_sec=sec_flavor' and '–mds_oss_sec=sec_flavor'.
- lmt should be able to add 'inter_mds_sec' and 'mds_oss_sec' into xml file and recognizable by lconf.

Servers will have options to accept only certain types of connections. When setup OSS/MDS via lconf, option "–deny-sec=sec_flavor[,sec_flavor...]" should be recognized and notify OSS/MDS kernel. Note for OSS, "–deny-sec" will deny the specified type of connection from both MDS and client. Currently we think this is good enough.

Maybe privacy connections to the OSS servers are only needed from the MDS, since there will be no secret transfer between OSS and client. And if we in the future support mixed security type in single security context, then integrity type might be enough for most cases. But anyway we provide the flexibility here.

3 Use Cases

3.1 Mount lustre at client

1. Sysadmin add options into config: lmt –mds_sec krb5p –oss_sec krb5i config.xml. And setup OSS/MDS ready.
2. User mount lustre by 'mount -t lustre server:/mds1/client /mnt/lustre'
3. Connections to MDS's are privacy protected, connections to OSS's are integrity protected.
4. User umount lustre.
5. User mount lustre by 'mount -t lustre -o mds_sec=krb5i,oss_sec=krb5p server:/mds1/client /mnt/lustre'
6. Connections to MDS's are integrity protected, connections to OSS's are privacy protected.
7. User umount lustre.
8. User mount lustre by 'mount -t lustre -o mds_sec=krb5p,oss_sec=krb5p server:/mds1/client /mnt/lustre
9. Connections to all MDS's and OSS's are privacy protected.

3.2 Startup MDS

1. Sysadmin add options into config: lmt –inter_mds_sec krb5p –mds_oss_sec krb5p config.xml
2. Sysadmin start mds by: lconf –node mds config.xml.
3. Connections between MDS's and MDS's to OSS's are privacy protected.
4. Sysadmin stop MDS's.
5. Sysadmin start mds again by: lconf –node mds –inter_mds_sec=krb5i –mds_oss_sec=krb5p config.xml.
6. Connections between MDS's are integrity protected, while MDS's to OSS's are privacy protected.

3.3 Deny certain type of connection

1. Sysadmin start OSS's by 'lconf –node ost1 –deny-sec=null config.xml'
2. Sysadmin start MDS's by 'lconf –node mds1 –mds_oss_sec=null config.xml', setup will fail because OST reject connection from MDS's.
3. Sysadmin start MDS's by 'lconf –node mds1 –deny-sec=null –mds_oss_sec=krb5i config.xml', will succeed.
4. Client mount by 'mount -t lustre -o mds_sec=null server:/mds1/client /mnt/lustre' or 'mount -t lustre -o oss_sec=null server:/mds1/client /mnt/lustre' will fail because either MDS's or OSS's will reject connection.
5. Client mount by 'mount -t lustre -o mds_sec=krb5i,oss_sec=krb5i server:/mds1/client /mnt/lustre' will succeed.

4 Logic Specification

With Kerberos, each service provider needs a service principal, and a corresponding service key installed. Usually the principal is bound to certain host for security. For example, currently lustre service principal is 'lustre/hostname@REALM'. While in clustered MDS case, we should use single principal for all MDS's, to minimize the administrator burden. It should be 'lustre@REALM' for all MDS's. Now we should break 'lustre@REALM' into 2 principals: 'lustre_mds@REALM' for MDS and 'lustre_oss@REALM' for OSS. All MDS's will be installed service key of 'lustre_mds@REALM', while all OSS's will be installed service key of 'lustre_oss@REALM'.

If MDS <=> MDS or MDS <=> OSS security is used, we also need start client gss daemon (lgssd) on MDS's at proper time. This needs to be incorporated into test scripts.

The interaction between kernel gss module and lgssd need some modification, which need to be notified the target service type (i.e. mds or oss) to issue the correct gss request.

The interface between kernel gss module and lsvcgssd also need some mofiication, because when OSS and MDS resident on a single node, lsvcgssd should be notified that using which service credential to verify the request.

Integrating security flavor setting into MDS's startup procedure and client's mount procedure needs to be integrated into the MDS startup configuration log.

Additionally the MDS and OSS should have configuration options that provide information on what kind of connections to accept.

5 State Management

MDS nodes need run lgssd if gss is active on any inter-server connections.

No disk format change. No special recovery consideration.

6 Alternatives

None.

7 Focus of Inspection

- Are there more clean design/divide on those new options?