Lustre* Security Today and in the Future

John Hammond

High Performance Data Division

April 2016

Lustre* security infrastructure has had some gaps

Uses standard POSIX file access control at filesystem level, like NFS

Has not previously had strong user or client authentication at the server

- Depends on network access control - address filtering or physical isolation
- Clients are trusted to provide accurate user identification

But... multiple industries require strict access controls to sensitive data

- Protected and personally-identifiable health information
- Classified data segregated based on multiple classification levels

New security capabilities are emerging

Existing and new security-related features are becoming available

- SELinux file labeling and Mandatory Access Control
- Nodemaps to classify clients by their NID with UID/GID maps per nodemap
- GSS/Kerberos node and user authentication and network encryption
- GSS/Shared Secret Key node authentication and network encryption
- Namespace subdirectory mounts

Security Component Overview

Authentication: Identification of an entity or principal

- *“I am who I say I am”* - e.g. Kerberos, SSK

Authorization: Grant access rights to an asset for authenticated principal

- *“I’m allowed to connect to this service”* – e.g. Kerberos, SSK
- *“I’m allowed to access these files”* – e.g. SELinux, UNIX permissions

Encryption: Help protect information from unauthorized interception

- *"I can safely communicate over this untrusted network"* - e.g. Kerberos, SSK

SELinux - Community 2.8, Enterprise Edition 3.0

Label files and enforce security policy based on process context

- Access enforced by kernel, users can't change policy
- Allow SELinux labels and policies to be set for all files
- Label stored in xattr, client open fetches label, caches it
- Allows enabling enforcement (MLS, RBAC) on *clients*

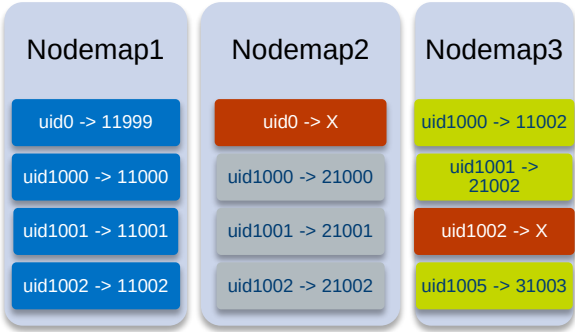
Limitations of current implementation

- No SELinux enforcement by *server* processes yet
  - Root compromise on client could disable SELinux
  - This is a significant problem beyond the scope of Lustre
  - Allow only trusted clients to connect using **Kerberos** or **SSK**
- Certification test cases not public, tied to specific hardware/software config

UID/GID Mapping - Community 2.8/2.9

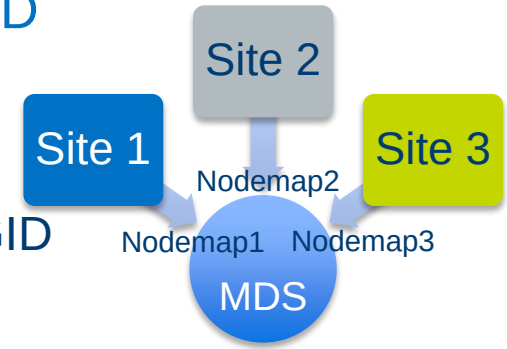
Identify clients from different administrative domains

- Group clients by NID into a *nodemap* on MDS/OSS
- Administrator sets *uidmap/gidmap* for each nodemap
- Link client to a specific nodemap at connection time



MDS and OSS map client's UID/GIDs to server UID/GID

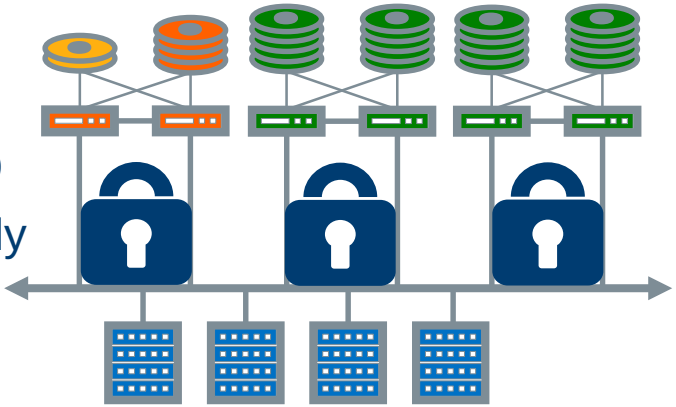
- UID/GID authentication on MDS, quota on MDSs/OSSs
- Map client root to non-0 UID or squash it, per nodemap
- Map unknown UID/GIDs to per-nodemap squash UID/GID
- Or reject access from unknown UID/GIDs completely
  - Only clients from that site can access owned by them



GSS/Kerberos and GSS/Shared Secret Keys

Different approaches for achieving similar goals

Kerberos - Community Edition 2.8, Intel® EE 3.0



- Authentication for nodes, and for users on the MDS
- Well known protocol, effective, but complex to setup
- Cross-realm authentication hard technically/politically

IU SSK - scheduled for Community Edition 2.9

- Self-contained in Lustre, easier to configure
- Crypto keys associated with a nodemap, positively identify clients
- Configure different keys for client/subnet/site, replace separately
- Uses kernel high-performance CPU offload of crypto calculations

Intel® Cloud Edition can use IPSec on TCP, but not with IB/OPA RDMA

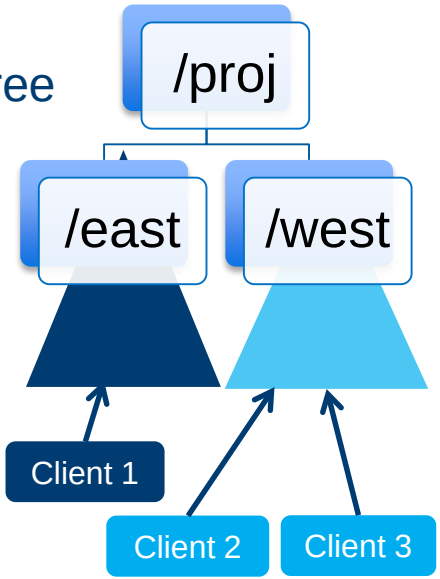
Subdirectory mounts/filesets - Community 2.9

Mount subdirectory of filesystem from MDS

- `mount -t lustre mgsname:/fsname/subdir /mnt/point`
- Client receives FID of subdir as root directory for mountpoint
- `.lustre` pseudo-directory not in subdirectory mount
- Client cannot look up names (`fid2path`) outside directory tree
- Clients normally share MDS and OSS resources
  - Unless admin isolates subtree to a dedicated DNE MDT

Not in itself a security feature, but can part of one

- Linked to nodemaps and secure client authentication
- Can provide isolated containers to identified clients



Security benefits of open development and review

All listed features openly developed and reviewed on *master* branch

- This leverages skills and perspectives from different organizations
- Increases pool of reviewers, more likely to find bugs
- Enhancements can target a wider variety of deployment scenarios

Good participation from multiple organizations

- UID/GID mapping, SSK design and code reviewed by several parties
- Testing of Kerberos and SELinux code by multiple vendors
- Review of Shared Key Crypto by Intel® internal security team ongoing

Growing developer community adds resources, interest, enhancements

Ongoing hardening of Lustre* code

Static analysis of entire code base for potential defects

- Several different tools being used - Coverity*, smatch, clang, and others
- Each one checks for different potential defects and security issues
- Some run on every patch, others run periodically on whole tree

Testing with fault injection to harden error handling paths and recovery

- Memory allocation errors, network message delay/loss

Dead code cleanup and simplification while adding several new features

- Fewer LOC = fewer defects, easier to understand and find higher-level bugs
- 2.5.0 to 2.8.0: 286k insertions - 257k deletions = net growth only +30kLOC

What is on the horizon?

Combining these features allows powerful new functionality

- Some final integration needed before they all work together seamlessly

Isolated subdirectory exports to clients => sub-filesystem containers

- Client auth + nodemap + subdir mount + rejection of unknown UID/GIDs
- SELinux + client auth to segregate files with different classification levels
- Allows multi-tenant cloud hosting, virtual environments, classified data

Need feedback from users and deployments for future directions

- Client-side data compression + encryption?
  - Per-user keys, secure on disk, safe erase, less overhead on server
- Need specific security threat model to determine gaps and priorities

Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit **http://www.intel.com/performance**.

Intel disclaims all express and implied warranties, including, without limitation, the implied warranties and merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing or usage in trade.

Intel, the Intel logo and others are trademarks of Intel Corporation in the U.S. and/or other countries.
*Other names and brands may be claimed as the property of others.


intel®
experience
what's inside™