Managing self-encrypting HDDs with Lustre/ZFS

LUG 2017

Q2-2017

Josh Judd, CTO
( Brief ) Agenda

• 15m, so this is an overview only – 10,000 foot view
• Full presentation will be at 2017 ORNL "Lustre Ecosystem”

Hanover Maryland, July 25-26
http://lustre.ornl.gov/ecosystem-2017

• Or, stop by WARP’s LUG booth to chat… er… pretty quickly…
Data at Rest Encryption

• Several options for encrypting disks
• An open software-only approach could be something like this:

```
cd /dev/disk/by-vdev
cryptsetup create eXXpAdYY eXXpAdYY
cryptsetup luksFormat /dev/mapper/eXXpAdYY
cryptsetup luksOpen eXXpAdYY eXXpAdYY
mkfs [ ... ] /dev/mapper/eXXpAdYY
```

**PROBLEM:** “Substantial” performance impact for SSDs (e.g. 50%) and any other CPU- or latency-intensive workloads

( **Note: e_p_d_ is WARP’s meaningful UDEV scheme for disk names )
Hardware Data at Rest Encryption

• **Solves** performance problems with software approach
• Historically, required expensive proprietary systems
• Now, can be done with **standard** hardware at low incremental cost
  – E.g., +2% or so system-level cost vs. equivalent non-encrypted drives

• **NEW PROBLEM**: Open software lags far behind for managing keys, lock states, and other encryption-specific features

• **New Solution**: DIY tools are not all that difficult to write
HGST TCG SAS Helium HDDs and SSDs

• Underlying hardware in reference solution: HGST TCG drives
• TCG = “Trusted Computing Group” standard for “Self Encrypting Drives”, which provides multiple benefits:
  – Transparency: No OS or app modifications required
  – Re-encryption: With SED, there is no need to ever re-encrypt data
  – Performance: No degradation in SED performance; hardware-based
  – Standardization: Whole industry is building to the TCG/SED specifications
  – Safety: Drives can be unlocked with multiple keys – can cancel keys known by one specific admin without effecting organization’s ability to access data
• BDE = “Bulk Data Encryption” is a similar standard, but strictly for lower end SATA drives, with fewer features and lower security
Open SED Functional Requirements

• At a high level, tools must handle cases such as:
  – Detect if a drive supports encryption, and if so, whether it is TCG vs. BDE
  – Manage PINs for all drives collectively and securely
    • Admins don’t have to manually unlock 1000s of drives in a single rack
    • PINs can be easily replicated and backed up
  – Turn drive locking on and off for individual drives or full systems
  – Allow all running directly-connected servers to “see” all drives, for HA
  – Allow drives to remain unlocked when OSS/MDS/MGS reboot or switchover
  – Manage PINs when replacing a failed drive
  – Handle lock status changes for re-seating drives
  – Display status of locking
Open SED Software Design

• WARP’s approach:
  – CLI utilities to manage – Encryption is changed very very rarely, and should not be changed by Jr Admins, so GUI management wasn’t a priority
  – Store the (large number of) drive PINs in a separate encrypted container file
  – Utilizes will accept a single password to unlock that file, then manage PINs on the drives for you
  – If you copy a single container file (backup or replicate) you’ll get all the PINs copied securely
  – If an admin quits, you can change just one password
Open SED “WARP Implementation” Walk Through

• Initial power on: All TCG drives are encrypting, but unlocked
  – They look just like any other drive, and are accessible to all attached servers
  – However, *internally*, they are already using 256-bit encryption
• Initialize drive locking with WARP’s “wmsedisk” tool
  – Creates encrypted “secure_keys_container” file, which contains all drive PINs, and can be backed up and/or copied to other WARP servers
  – All drives are now encrypted *and* protected, so that they would be unreadable if powered off
  – However, they are currently unlocked and thus visible to all directly SAS-attached WARP servers
• Create pools and filesystems, *if* they didn’t already exist
  – Unlike software encryption method, this step actually *can* be performed *first*
Open SED Walk Through (continued)

• Test to ensure locking is working as expected
  – Completely power down all servers and JBODs
  – Power on servers then JBODs
  – All drives should be locked and *not usable* by any of the servers

• Log into any attached server which has “secure_keys_container”

• Send command to unlock all drives with “wmsedisk”
  – Prompts for your PIN container password, and makes drive PINs available
  – The server you’re on will now see all drives as mountable
  – Run “partprobe” on all other directly-attached servers to get them to notice

• Import all zpools to their associated servers, and start Lustre

• Until the next cold boot of the JBODs, or ejection of HDDs/SSDs, it should work like any other Lustre system
Questions?
Please stop by WARP/HGST booth

Q2-2017
Josh Judd, CTO