



ORNL / DoD Lustre Ecosystem 2016

Open Source Lustre Security

Q1-2016
Josh Judd, CTO



Safe Harbor

This presentation is considered confidential, and is covered under the terms of the Non-Disclosure Agreement (NDA) in place between WARP Mechanics Ltd and the attendees viewing the content.

This document is supplied “AS IS”, for information only, without warranty of any kind, expressed or implied. WARP Mechanics reserves the right to change this document at any time, without notice.

NOTHING IN THIS PRESENTATION SHALL CREATE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS, WITH RESPECT TO ANY PRODUCTS AND/OR SERVICES REFERENCED HEREIN.

Any or all of the products in this presentation may be under development at this time. Specifications, such as, without limitation, release dates, prices, and product features, may change. The products may not function as described, and a production version of the products may never be released. Even if a production version is released, it may materially differ from the version discussed in this presentation.

WARP Mechanics, WARPware, the WARP Mechanics logo, the WARP Mechanics icon, and SmartStorage System are trademarks of WARP Mechanics Ltd. in the United States and other countries. All other brand, product, or service names may be trademarks or service marks of, and are used to identify, products or services of their respective owners.

Agenda

- Overview
- Physical security
- Securing Linux
- Data at rest encryption
- Lustre white list / black list example
- Call to action

Note: Suggest detailed questions at lunch or via email

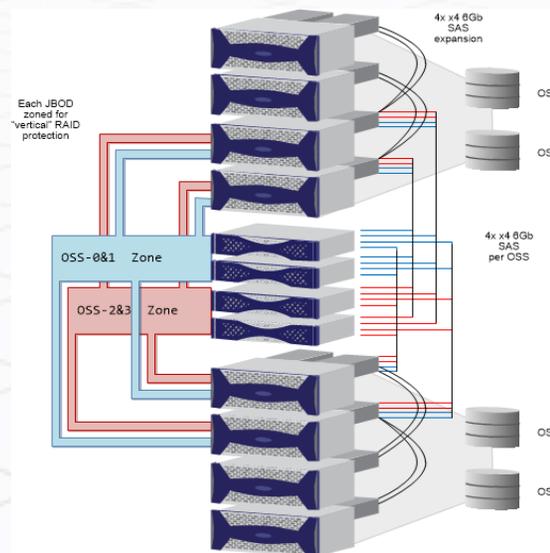
Caveats etc

- WARP *only* does ZFS solutions
 - “x” over ZFS
 - Comprehensive set of ZoL enhancements and tools
 - ZFS₂ architecture
 - Basically, we’re the “go to” guys for commercially supportable ZoL...
- Therefore we have done nothing with ldiskfs (since 2011)
- Testing for WARP’s Open Secure Lustre recommendations has been done on WARP’s hardware and OS, not on any other platform

Test Scale Systems

- WARP has a number of PB scale test systems
- These are not 10s of PB, but representative of 1x SSU
- For example, security processes were tested on this example SSU before it went production:

- 8x high density JBODs
- Connected to 4x ZFS OSSs
- Separate HA ZFS MDS/MGS
- Running 2x Lustre FSs
- Uses SSD/HDD hybrid
- Planning to test in larger scale systems next month



2PB to 4PB Lustre/ZFS WARP SSU



Overview

- “Secure” is not yes/no – no system is either secure or non-secure
- Anything that *can* be accessed can theoretically be hacked
- Anything that *cannot* be accessed is useless as a parallel FS
- Therefore “Secure Lustre” *must* be a balancing act
- This lecture discusses how WARP believes a “sufficiently secure” Lustre can be achieved with zero proprietary software
- Our balance formula is:
 - No “vendor lock” allowed
 - Reasonably easy to implement
 - Reasonably low performance impact
 - Reasonably useful improvement to security

Physical Security

- Axiom: If the bad guy can get to the hardware, you already lost
- Something you have, something you know, something you are
- Cabinet and cage locks
- Every inch of cable
- Every terminal allowed to log into system
- Camera... Which actually store footage
- Literal back doors

Linux Security

- Often, Lustre servers are “the exception” to “normal” security
 - SE Linux off, IP tables off, etc.
- *Might* be valid, up to a point...
- But if somebody can hack the OS, does securing *Lustre* help?
- Example: One WARP customer wanted “enhanced” Lustre security, but had literally not even changed default passwords
- In short, do the basic stuff first
- E.g., no SUID/SGID bits allowed on FS

SSH and other services

- Change default port for ssh

```
vi /etc/ssh/sshd_config
```

```
→ Port 40122
```

- Disable **all** services that you **aren't** using

```
chkconfig smb off ; chkconfig nfs off ; chkconfig fcoe off ...
```

IP Tables with Lustre

- At minimum, set it up on the MGS
 - Maximum effect with minimum performance overhead
 - If a client tries to connect outside of correct IP range, MGS won't talk to it

```
iptables -P INPUT ACCEPT ; iptables -F ; iptables -A INPUT -i lo -j ACCEPT  
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

change above to correct SSH port

```
iptables -P INPUT DROP ; iptables -P FORWARD DROP ; iptables -P OUTPUT ACCEPT  
iptables -L -v
```

```
chkconfig iptables on  
service iptables start
```

SE Linux with Lustre

- Since Lustre 2.3, SELinux can work with Lustre
- BUT, has noticeable performance impact as well as admin overhead
 - E.g., could reduce performance by 50% on typical workloads
 - Even higher for “ls -l” type workloads
- May have minimal benefit, so may be more trouble than it’s worth
- If you want to go there... Set “permissive” & reboot, see what’s happening; adjust

```
# grep -i "SELinux is preventing" /var/log/messages
```

```
Mar  7 14:52:19 WARPhpc-658-RC1 setroubleshoot: SELinux is preventing  
/bin/bash from read access on the lnk_file /etc/sysconfig/network-  
scripts/ifcfg-eth0. For complete SELinux messages run sealert -l  
2ecf8ed8-3608-4c07-9d5a-e687d477ca10
```

Account and Password Policies

- Change root password – we see default passwords on “appliances”
- Limit sudo and don't log in directly as root
- Disable all local user-level accounts for log in
- WARP can support 100% diskless OSS/MDS/MGS – centralizes all account security, right? Still need to remember IPMI accounts
- Anything with clear text IPMI password needs to be locked down
- Look at [/etc/login.defs](#) and [/etc/pam.d/system-auth](#) for:
 - Password Aging
 - Password Length
 - Password Complexity
 - Number of Login Failures
 - Re-Used Password Deny

Data at Rest Encryption

- Several options for encrypting disks
- Plenty involve replacing disks etc, but there's also this:

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

“Substantial” performance impact for SSDs, e.g. 50%

(*Note: e_p_d_ is WARP's meaningful UDEV scheme for disk names*)

Lustre White List / Black List

- Credit: Feature funded by Naval Research Lab (NRL)
- Jeremy Filizetti (ultrascale.net) created a white list:
 - review.whamcloud.com/#/c/18672
- Assume you already have appropriate Lustre server kernel
- Git 2.7 or 2.8 source, and apply patch

```
git clone git://git.whamcloud.com/fs/lustre-release.git
cd lustre-release
git fetch http://review.whamcloud.com/fs/lustre-release
refs/changes/72/18672/1 && git cherry-pick FETCH_HEAD
```
- Make patched Lustre RPMs

```
sh autogen.sh ; sh configure && ( make && make rpms )
```

Lustre White List / Black List *(cont.)*

```
# lctl get_permitted_nids  
ALL
```

```
# lctl list_nids  
10.0.0.221@tcp
```

```
# lctl set_permitted_nids 10.0.0.221@tcp
```

```
# lctl get_permitted_nids  
10.0.0.221@tcp
```

NID range format is same as root squash; supports "NONE" and "ALL" as well

Lustre White List / Black List *(cont.)*

- Does not implement “black” list expressly
 - However, white list function implies black list function
 - E.g., say you specify NID range 192.168.1.0/24
 - You want to “knock out 192.168.1.100 temporarily
 - Change white list to 192.168.1.1-99 + 192.168.1.101-254
 - Less efficient for sure... But...
-
- 1. Change range
 - 2. Send “offending” NID to
`/proc/fs/lustre/obdfilter/*OST*/evict_client`

Security Related Sites and Lists

www.tldp.org/HOWTO/Security-HOWTO

en.wikipedia.org/wiki/Security_Technical_Implementation_Guide

lists.centos.org/mailman/listinfo/centos-announce

access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html

Send a note to lustre@warpmech.com, if you want more specific security examples

Calls to Action

“Live free or insert perpetual vendor lock name here”

- #1: Keep it free
- #2: Keep it free...
- Send a note if you have an environment to do higher scale tests
 - lustre@warpmech.com
- Review all *basic* Linux and Network security for Lustre servers
- At minimum, implement the ones that don't reduce performance
- At least evaluate the ones that do reduce performance
 - Adding OSSs may be cheaper than being hacked



ORNL / DoD Lustre Ecosystem 2016

Open Source Lustre Security

Q1-2016
Josh Judd, CTO

