Driver for Linux Release Notes

Versions: Driver for Linux, FC/FCoE Version 8.3.7.33 (RHEL 6.4–6.5 / SLES 11 SP2, SP3)
Driver for Linux, FC/FCoE Version 8.2.2.30 (RHEL 5.9–5.10 / SLES 10 SP3, SP4)
Driver for Linux, NIC Version 4.9.230.0
Driver for Linux, iSCSI Version 4.9.225.0 (RHEL 6.4–6.5 / SLES 11 SP2, SP3)
Driver for Linux, iSCSI Version 4.9.196.1003 (RHEL 5.9–5.10 / SLES 10 SP3, SP4)

Date: October 2013

Purpose and Contact Information

These release notes describe the resolved known issues and current known issues associated with these Emulex® driver for Linux releases.

For the latest product documentation, go to www.Emulex.com. If you have questions or require additional information, contact an authorized Emulex technical support representative at tech.support@emulex.com, 800-854-7112 (US/Canada toll free), +1 714-885-3402 (US/International), or +44 1189-772929 (Europe, Middle East, and Africa).

Resolved Issues

Linux FC/FCoE Versions 8.3.7.33 and 8.2.2.30 Resolved Issues

1. **Added support for Emulex LightPulse® LPe16202, LPe16004, and LPe15004 adapters.**

   Note: Drivers and applications may report the LPe16202 adapter as follows, depending on the mode in which the adapter is running:

   - FC mode: LPe16202
   - FCoE mode: OCe15100 FCoE

2. **Corrected an issue with multipath hanging for Test Unit Readys (TURs) with large timeouts.**
3. **Corrected an issue with an incorrect FCP Priority value being detected for LUNs greater than 128.**
4. **Corrected an issue where a PCI function reset could not be performed when the adapter was in offline mode.**
5. **Corrected a potential failure when setting the SLI-3 adapter mode for performing certain OneCommand™ Manager application commands.**
6. **Corrected an issue with SLI-3 failing an FCP write due to an FCP response of "Check Condition; No Sense" with a residual zero.**
7. **Corrected the maximum value for lpfc_lun_queue_depth to be 512.**
8. **Corrected an issue with the Receive Queue and varied frame size handling.**
9. **Corrected an issue that prevented the OneCommand Manager application to correctly display the Service Processor and ULP firmware names on Power PC (PPC) systems.**

10. **Corrected an issue with the mailbox wait routine failing to issue a dump memory mailbox command.**
11. **Clarified the behavior of the lpfc_max_luns module parameter.**
12. **Corrected an issue with the FCoE connection list record using an invalid VLAN identifier on PPC systems.**
13. **Corrected an issue that prevented the OneCommand Manager application from reporting Forward Error Correction (FEC) status.**
14. **Corrected an issue where a devloss timeout race condition resulted in a null pointer reference causing a system lock up.**
15. **Corrected an issue with SCSI target bus reset handlers not returning error status due to a port timeout.**
16. **Corrected an issue with a potential system panic while handling an unsolicited receive buffer error condition.**
17. **Corrected a potential system lock up during FCoE failover testing.**

Linux FC/FCoE Version 8.3.7.33 Resolved Issues

There are no resolved issues specific to this version for this release.

Linux FC/FCoE Version 8.2.2.30 Resolved Issues

There are no resolved issues specific to this version for this release.

Linux NIC Resolved Issues

1. **Added support for the LPe16202 converged fabric adapter (CFA).**

   Note: The NIC driver and applications may report the LPe16202 adapter as OCe15100 NIC.

2. **Corrected an issue with PCIe extended error handling (EEH) when a PCIe error was detected in succession.**
3. **Corrected an issue with error recovery failing on LPe16202 adapters.**
4. **Corrected an issue that occurred during EEH-based error recovery where the NIC driver would exit with a firmware POST timeout.**
5. **Corrected an issue with EEH recovery failing because the driver failed to load due to improper initialization attempts.**
6. **Corrected an issue that potentially could have caused a hardware error.**
7. **Corrected an issue where setting up virtual functions (VFs) through a physical function (PF) would fail on certain Linux kernels (3.8 and later).**
8. **The error log displays the actual supported VFs.**
9. **Corrected an issue with firmware download for LPe16202 adapters causing Ethernet devices to disappear on RHEL 5.9 systems.**

10. **Corrected an issue where the NIC adapter fails to initialize after a suspend-resume cycle.**
11. **Corrected an issue with the adapter logging an "opcode 59 1 failure with status 1 76" error due to a MAC collision when the bond driver attempts to change the primary interface.**
12. **Corrected an issue when the controller moves to multicast promiscuous mode by adding the following log message – "Exhausted multicast HW filters. Disabling HW multicast filtering."**
13. **Corrected an issue where VF interfaces assigned to the VM lose the MAC address after rebooting the VM.**
14. **Corrected an issue where the driver could lead to a firmware hang when "ifup ethx" and "ifdown ethx" are run in a loop.**
15. **Corrected an issue with an intermittent system crash occurring when "ip link set down eth2" and "ip link set up eth2" are run in a loop.**
16. **Corrected an issue with low, bi-directional throughput on 1G ports of OCe11000-series four-port adapters.**
17. **Added support for a statistic counter that tracks dropped packets in the transmit path.**

Linux iSCSI Version 4.9.225.0 and 4.9.196.1003 Resolved Issues

1. **Previously, Emulex provided a proprietary iSCSI driver as the out-of-box driver for all Linux operating systems. The Open-iSCSI driver was available inbox only. For RHEL 6.4 and SLES 11 SP2 and later drivers, Emulex is transitioning the Open-iSCSI driver to be the out-of-box driver. For RHEL 5.x and SLES 10 SPx, the out-of-box driver will continue to be the proprietary driver.**

Linux iSCSI Version 4.9.225.0 Resolved Issues

There are no resolved issues specific to this version for this release.

Linux iSCSI Version 4.9.196.1003 Resolved Issues

There are no resolved issues specific to this version for this release.

Known Issues

Linux FC/FCoE Versions 8.3.7.33 and 8.2.2.30 Known Issues

1. **Link Aggregation Control Protocol (LACP) cannot be used on the same port as FCoE or iSCSI.**

   Workaround

   None.

2. **PCI Hot Plug may cause applications such as the OneCommand Manager application or third party applications that use the Emulex libraries (for example, HBAAPI), to misbehave or malfunction.**

Workaround

a) Stop all applications that are accessing LPFC's HBAAPI interface (OneCommand Manager application or third party applications) before performing PCI Hot Plug of an LPFC adapter.

b) Use the following command to stop the OneCommand Manager application:

```
#/usr/sbin/hbanyware/stop_ocmanager
```

c) After performing PCI Hot Plug of the adapter, you can restart the applications.

3. **Deletion of Vports/PCI Hot Unplug**

On occasion the kernel might report SCSI errors when deleting Vports via the sysfs interface or performing a PCI Hot Unplug of an Emulex adapter:

```
kernel: Synchronizing SCSI cache for disk
kernel: FAILED
```

-or-

```
SCSI error: return code = 0x00010000
```

Workaround

None.

These messages do not indicate a functional failure and can be ignored.

4. **Deleting Vports while devices are in use.**

Emulex provides management utilities that allow you to delete Vports. However, there is no mechanism for the LPFC driver to detect whether devices accessed through that Vport are in use. This means you can delete a Vport when devices accessible through the Vport are mounted or when I/O is outstanding to the device. When file systems are mounted on Vports and Vports are deleted, the file systems still appear to be mounted; however, they be inaccessible.

Workaround

Before deleting Vports you must prepare the system affected by the Vport deletion accordingly, by unmounting all the devices accessible through the Vports, and ensuring there is no outstanding I/O.

5. **Support of 4Gb/s adapters in Direct I/O virtualized environments.**

Default driver configuration fails to initialize 4 Gb/s adapters in virtualized environments that use Direct I/O or SFPT. This may result in a system hang or an uninitialized LPFC adapter in Intel VT-d and AMD-V IOMMU systems.

Workaround

In these virtualized environments that use Direct I/O or SFPT and 4 Gb/s adapters, you must load the LPFC driver with the following driver parameters set:

- lpfc_hostmem_hgp=1
- lpfc_sli_mode=2

For example:

```
# modprobe lpfc lpfc_hostmem=1 lpfc_sli_mode=2
```

Note: A consequence of this resolution is that virtual ports are no longer supported by the LPFC driver when the lpfc_sli_mode parameter is set to 2.

6. **Order of LPFC module in the initrd module list.**

   On SLES 10 and SLES 11 systems, if another SCSI driver, such as aic79xx, is loaded right behind the LPFC driver through the initrd image, the loading of the SCSI driver right after the LPFC driver might interrupt the SCSI mid-layer discovery process on the LUNs connected to the LPFC driver. This causes the SCSI discovery reference count put on the LPFC driver to not release and the LPFC driver cannot unload.

   Workaround

   The Emulex driver kit installation script always puts the LPFC module at the end of the INITRD_MODULES list before building the initrd image. However, if this problem is observed make sure no SCSI drivers are added right after the LPFC module in the INITRD_MODULES list.

7. **The physical interface can improperly name eth0.123 when the /etc/sysconfig/network-scripts/ifcfg-eth0.123 file contains a HWADDR declaration.**

   When using VLAN on RHEL 6.x, the main interface is not created, and the VLAN interface does not actually work correctly as a VLAN.

   This is an issue with the udev script in RHEL 6.x. In RHEL 6.x, the udev scripts interpret the HWADDR field in an ifcfg-ethX.Y configuration file to mean that the configuration belongs to a real interface. It does not check for the existence of the VLAN field to exclude the field as a real interface. Therefore, you may see unpredictable behavior when including the HWADDR field in the ifcfg-ethX.Y file for a VLAN. The setting may, or may not, work. When this setting does not work, it is because the scripts have created or renamed the main interface as ethX.Y instead of just ethX. Later, when adding the VLAN interface, it fails because ethX does not exist.

   Workaround

   Remove the HWADDR line in ifcfg-ethX.Y files that refer to VLAN configurations. You must also clean up the /etc/udev/rules.d/70-persistent-net.rules file to remove all the incorrect references to ethX.Y.

   For more information, see https://bugzilla.redhat.com/show_bug.cgi?id=723936.

8. **The 16Gb LPe16000B and LPe16002B adapters are not supported with the RHEL 5.9 and RHEL 6.4 inbox FC/FCoE drivers.**

   The RHEL 5.9 inbox FC/FCoE (LPFC) driver version 8.2.0.128.3p and the RHEL 6.4 inbox FC/FCoE (LPFC) driver version 8.3.5.86.1p do not support the 16Gb LPe16000B and LPe16002B adapters (PCI Express 3.0 Bus Type). However, these driver versions do support the 16Gb LPe16000 and LPe16002 adapters (PCI Express 2.0 Bus Type).

   In Boot from Storage Area Network (SAN) configurations over the unsupported LPe16000B and LPe16002B adapters, the process of booting from SAN will fail. In standard local disk boot configurations, although the FC/FCoE driver will load upon system boot, the target storage LUNs will not be discovered.

   Note: You can use the Linux “lspci” utility to determine whether an adapter is LPe1600xB or LPe1600x, and therefore whether it is supported by the RHEL 5.9 inbox LPFC driver version 8.2.0.128.3p or the RHEL 6.4 inbox LPFC driver version 8.3.5.86.1p.

   Executing the “lspci -x” Linux utility displays the PCI configuration space for each device in the form of:

```
<bus/slot/function info> <strings matching vendor id, etc>
00: xx xx xx xx ...
10: xx xx xx xx ...
…
```

For example, for the Emulex 16Gb adapter:

```
0a:00.1 Fibre Channel: Emulex Corporation Lancer-X: LightPulse
Fibre Channel Host Adapter
00: df 10 00 e2 46 00 10 00 00 00 04 0c 10 00 80 00
10: 0c 00 cd f1 00 00 00 00 00 00 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 df 10 00 e2
30: 00 00 00 00 54 00 00 00 00 00 00 00 0b 02 00 00
```

The 9th byte (offset 0x8) of the PCI configuration space of the LPe1600x/LPe1600xB adapters indicates the Revision ID register value of the adapter.

- If the Revision ID register value is 0x00, this represents an LPe1600x adapter.
- If the Revision ID register value is 0x10, this represents an LPe1600xB adapter.

In the example above, the value of the byte with offset 0x8 is 0x00; therefore, this represents an LPe1600x adapter, so the RHEL 5.9 inbox FC/FCoE (LPFC) driver version 8.2.0.128.3p and RHEL 6.4 inbox FC/FCoE (LPFC) driver version 8.3.5.86.1p do support this adapter.

Workaround

Support for the 16Gb LPe16000B and LPe16002B adapters will be provided with the FC/FCoE (LPFC) driver included in the RHEL 6.5 release. In addition, support is provided for the RHEL 5.9 and later releases through an independent out-of-box FC/FCoE (LPFC) driver packages hosted on the Emulex website.

Linux FC/FCoE Version 8.3.7.33 Known Issues

1. **Devloss timeout after swapping ports.**

   The driver may not finish discovery when two initiator ports are swapped. This causes all devices accessible through one or both of these initiator ports to time out and all I/O to fail.

   Workaround

   Do one of the following:

   - When swapping cables, replace each cable, one at a time, and allow discovery to finish before replacing the next cable. To determine if discovery is finished, read the “state” sysfs parameter.
   - When swapping cables, allow devloss timeout to fire before replacing the cables. (This fails all outstanding I/O.)

2. **Driver package installation failure may occur when installing on systems with SLES11 SP2 maintenance kernel version 3.0.80-0.7.**

   The SLES11 SP2 maintenance kernel release with version 3.0.80-0.7 has a kernel version higher than the latest SLES11 SP3 GM release (which was 3.0.76-0.11.1). This is a non-standard kernel versioning scheme that is used by Linux OS vendors. See the related SuSE KB article at https://www.suse.com/support/kb/doc.php?id=7012717.

This non-standard versioning scheme causes the Emulex driver package installation scripts to fail when an attempt is made to install the packages on this specific SLES11 SP2 maintenance release.

Workaround

To install the driver on this specific SLES11 SP2 maintenance kernel (version 3.0.80-0.7), you should not use the driver package's installation script. Instead, you should manually install the driver binary RPM built for the SLES11 SP2 GM kernel (version 3.0.13-0.27) and corresponds to the architecture of the target system (for example, x86_64, PPC, and so on). These driver binary RPMs include the kernel version (3.0.13_0.27) in their name.

For example:

```
# rpm -Uhv
elx-lpfc-kmp-default-8.3.7.27_3.0.13_0.27-1.sles11sp2.x86_64.rpm
```

Linux FC/FCoE Version 8.2.2.30 Known Issues

1. **LILO Boot Loader is not supported on i386 and x86_64 architectures.**

   The LILO Boot Loader on i386 and x86_64 architectures is not supported for this driver. If the LILO boot loader is used, after the LPFC driver package is installed and upon reboot an incorrect initial ramdisk is used, and the system might not boot correctly.

   Workaround

   The boot loader supported with this driver is GRUB, which is the default boot loader for most of the Linux distributions. LILO is an older boot loader used on i386 and x86_64 architectures only. GRUB works correctly with the driver package's installation script.

2. **Suspend to disk command results in a kernel Oops.**

   If you attempt to suspend to disk using the command:

   ```
   #echo disk > /sys/power/state
   ```

   the LPFC driver encounters a kernel Oops.

   The sysfs parameter "/sys/power/state" is used to suspend and resume the system. The LPFC driver does not support the suspend to disk and resume command. Do not attempt to use this sysfs parameter when the LPFC driver is loaded.

   Workaround

   None.

3. **Potential error messages during the driver kit removal process.**

   As part of the driver kit removal process initiated via the "lpfc-install -u" command the previous in-box LPFC driver version (driver version part of the Linux distribution), which was saved as part of the current driver kit install process, is restored and becomes the active driver. However, the driver kit un-installation process by design does not remove any entries in the Linux distribution configuration file (modprobe.conf). As such, parameters that would have been valid for the just-removed driver versions and entered in the modprobe.conf file, are also used to load the just-restored in-box driver version. This potentially create problems when:

- The just-removed driver version might include module parameters that did not exist in the older just-restored driver version, and
- One or more of these module parameters are included in the configuration file (modprobe.conf).

If the above criteria are met, you see an error message during the uninstallation process of the driver kit, such as:

```
Loading LPFC Driver .FATAL: Error inserting lpfc
(/lib/modules/<kernel_revision>/kernel/drivers/scsi/lpfc/lpfc.ko):
Unknown symbol in module, or unknown parameter (see dmesg)
```

For example, this issue can be observed is when uninstalling an 8.2.0.x driver kit, which had DH-CHAP functionality enabled, on a Linux distribution with an older 8.1.10.x in-box driver version.

Workaround

If such an error is seen during the kit removal process, you need to edit the Linux configuration file (modprobe.conf) and remove all entries that list LPFC driver module parameters, that is entries that start with:

```
options lpfc ...
```

Then attempt to uninstall the driver kit again.

Note: To find the module parameters supported by an LPFC driver module, type:

```
# modinfo <driver_dir>/lpfc.ko
```

4. **Potential connection loss due to an FCF Failover issue with a Cisco FCoE switch.**

An issue was discovered with Cisco Nexus 5000-series FCoE switch firmware 4.1(3)N2(1) or earlier, in NPV mode, that may cause the loss or interruption of SCSI connections when used with the Emulex OneConnect UCNAs. The switch incorrectly sends out a Discovery Advertisement to All-ENode-MACs from the FCF MAC with which the FC uplink was down. The end result is that sometimes the UCNA hangs on to an offline FCF or experiences back-to-back FCF failover and it may potentially lead to a Linux SCSI mid-layer devloss timeout.

Workaround

It has been verified that Cisco's 4.2(1)N1(1) release has corrected the issue. Emulex highly recommends that you upgrade your Cisco Nexus 5000-series FCoE switch firmware to 4.2(1)N1(1) or later to avoid this issue. If you decide to use 4.1(3)N2(1) or earlier firmware with your Cisco Nexus 5000-series FCoE switch and this issue is encountered, increase the FC transport dev_loss_tmo parameter to 60 seconds. This can be accomplished in one of two ways:

- Update the FC transport dev_loss_tmo parameter. For example:

```
# echo 60 > /sys/class/fc_remote_ports/rport-3:0-1/dev_loss_tmo
```

-or-

- Update the LPFC driver's lpfc_nodev_tmo parameter. For example:

```
# echo 60 > /sys/class/scsi_host/host3/lpfc_nodev_tmo
```

5. **Inband management connection loss during FCF failover with Brocade FC switch.**

   During FCF failover from one FC uplink to another in configurations with a Cisco FCoE switch in NPV mode and a Brocade DCX Director FC switch with firmware 6.1.1a, there can be cases where the OneConnect UCNA symbolic node names of the FCoE UCNA interfaces involved in the FCF failover disappear. This is observed from the OneCommand Management application, where the inband communication from one of the UCNAs to the other involved in the FCF failover is lost.

   During the FCF failover from one FC uplink to another, although all the OneConnect UCNAs failed over successfully for fabric logging and target rediscovery, it was observed that occasionally the Brocade DCX Director FC switch failed to establish the initiator-to-initiator forwarding with the UCNAs involved in the failover. This is based on the requirement that the FC switch should forward the initiator-to-initiator I/Os received from the FCoE uplink back into the same FCoE uplink. This issue has been observed with the Brocade DCX switch running firmware version 6.1.1a.

   Workaround

   None.

Linux NIC Known Issues

1. **ifconfig displays “RUNNING” flag when the cable is not connected.**

   This is a known issue in the Linux operating system stack.

   Workaround

   Use “ip link show” to properly display the “Oper status” value.

2. **On SLES 11 SP2 (or earlier) systems, if the NIC driver is unloaded in a hypervisor when VFs are assigned to a guest operating system, unexpected behavior may result.**

   Workaround

   To move to a different NIC driver or load the driver without enabling VFs (num_vfs=0), reboot the host.

3. **Kdump is not supported when SR-IOV is enabled.**

   Workaround

   None.

4. **In certain skews of controllers, the initialization of the 30th VF fails.**

   Workaround

   None.

5. **PING is not working when attempting to bridge the 1G or 10G ports to the virtual machines when SR-IOV is enabled for 10G ports in the BIOS.**

   This issue occurs due to limitations of the virtual Ethernet bridge. All transmitted broadcast packets are looped back by the controller. This affects the functionality of the Linux bridge, as it appears as if the same ARP broadcast packets are received on two different interfaces.

   Workaround

   1) Set the aging of the bridge to 0 using the following command:

```
"brctl setageing <bridge> 0"
```

This causes the bridge to behave like a hub and flood the packet to all the ports (except the one on which the packet arrived) every time. This may impact performance. If you have only two interfaces on this bridge (one NIC interface and virbr0-nic), there is no performance impact.

2) Another option may be to use the MacVTap interface to the guest instead of the bridge interface. The MacVTap interface is only available in the RHEL 6.3 or newer kernels.

6. **Unable to ping the RHEL 6.4 KVM guest (VM) from a remote host when using the MacVTap driver and IPv6.**

   When an IPV6 unicast address is configured, the Linux kernel also configures an Ethernet multicast MAC address in the format 33-33-xx-xx-xx-xx. In the context of a guest operating system (VM), the mcast address that needs to be configured comes into existence after the VM is booted up and the unicast IPv6 address is configured in the guest operating system.

   When using a MacVTap interface to bridge the NIC interface to the VM's interface, the previously-mentioned Ethernet MAC address configuration in the VM does not reach the MacVTap interface automatically. Since the MacVTap interface does not put the NIC interface into promiscuous mode, this results in an IPv6 ping failure from the remote host to the VM's interface.

   Workaround

   Select one of the following workaround options:

   - Use the following command:

     ```
     ifconfig eth<x> allmulti
     ```

     where "eth<x>" is the interface in the hypervisor to which the MacVTap interface is bridged. However, this causes all multicast traffic to be received.
   - Alternatively, you can configure the unicast IPv6 address (assigned to the VM interface) on the MacVTap interface in the hypervisor. This results in the Ethernet mcast MAC address being configured in the NIC interface.

     Use the following command:

     ```
     ifconfig macvtap<x> inet6 add <ipv6 address>
     ```

     where "macvtap<x>" is the MacVTap interface in the hypervisor and "<ipv6 address>" is the unicast IPv6 address assigned to the VM interface.

7. **When SR-IOV is enabled, NIC priority group (PG) and priority flow control (PFC) are not supported.**

   Workaround

   None.

8. **In certain skews of controllers, when the driver is loaded with num_vfs=32, the initialization of two of the VFs fails and only 30 VF interfaces are created.**

   Workaround

   None.

9. **In certain configurations, timeout errors may occur during maximum performance (MPx) socket testing.**

   Workaround

   For Linux Xen:

   - Only configure interfaces created with the "netfront" source model and ignore the second set of interfaces created with "8139cp"

   For Linux KVM:

   - Use the "virtio" device model instead of "Hypervisor Default"

10. **When using bonding on top of VLANS on a RHEL-based system, networking appears to hang during system boot or when restarting the network.**

    On a RHEL-based system, having a bond on top of a VLAN while using ONBOOT=yes, can cause the system to hang during boot or when restarting the network. The system will appear to hang while trying to start one of the slave interfaces.

    For example, if you have a configuration similar to the following:

```
DEVICE=bond0
BROADCAST=10.255.255.255
GATEWAY=10.0.0.254
IPADDR=10.0.0.14
NETMASK=255.0.0.0
ONBOOT=yes

BONDING_OPTS="mode=0 miimon=100"
DEVICE=eth2.2
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
VLAN=yes

DEVICE=eth3.2
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
VLAN=yes
```

    This system hangs because the RHEL networking scripts try to bring up the eth2.2 and eth3.2 interfaces twice. It will bring them up once because the bonding interface specifies them as slave interfaces, and then it tries to bring them up a second time because it sees the ONBOOT=yes parameter on a VLAN device.

The issue occurs because after the devices are first brought up, the MAC addresses of the interfaces changes. Usually the first interface's MAC address is set on the rest of the interfaces. When the device is brought up a second time, the MAC address of the interface no longer matches the HWADDR field. The network scripts will wait for an interface with a matching MAC address to appear, which does not occur.

Workaround

Set the ONBOOT parameter to “no” on the slave interfaces. This prevents the system network scripts from trying to bring up the interface a second time. This will not negatively affect the operation of the bonded interface because the ifcfg-bond0 file still has ONBOOT set to yes. The networking scripts will attempt to bring up the bond0 interface on boot, which will cause the slave interfaces to be brought up and configured correctly.

If you are only using bonding without VLAN, the network scripts do not bring up slave interfaces, so the issue is avoided. The issue occurs only when bonding is on top of VLAN interfaces.

Linux iSCSI Version 4.9.225.0 Known Issues

1. **The limitations specified in Appendix A of the *Emulex Drivers for Linux User Manual* also apply to the out-of-box Open-iSCSI driver.**

   Workaround

   None.

Linux iSCSI Version 4.9.196.1003 Known Issues

There are no known issues for this release.

Technical Tips

1. On systems running RHEL 6.4 or later, when upgrading or downgrading from the proprietary be2iscsi driver to the open be2iscsi driver, you must install the Dracut-network RPM package, and pass the netroot command parameter into Dracut to build the initramfs image.
2. To update the proprietary iSCSI driver to the Open-iSCSI driver for SLES 11 SP2 and later, you must do the following:
   a) Make sure the Open-iSCSI package is installed. If it is not, mount the installation CD and install it.
   b) Edit /etc/iscsi/initiatorname.iscsi to add the initiator IQN name. This name can be the same name as in iSCSISelect or a new unique one.

      Note: The name must conform to the standard IQN.

   c) (Optional step): Change directory to /boot , copy the default initrd to a new name, and create a boot entry in /boot/grub/menu.lst for this new initrd name using the same kernel. This step ensures that the system can still boot if something goes wrong.
   d) Install the Emulex Open-iSCSI driver .rpm file.

e) If this not an iSCSI boot system, reboot and follow the Open-iSCSI instructions to log into targets; you do not need to continue following these steps.

   If this is an iSCSI boot system that previously used the proprietary iSCSI driver, do not reboot. Instead, continue with the next step.

f) Change directory to /boot, rebuild the initrd to include the "iscsi" feature. The output of this command lists the capabilities included in the initrd. Make sure "iscsi" is listed. This option includes capabilities such as iscsiadm in the initrd.

   For example: linux-fiai:/boot # mkinitrd -i initrd-3.0.76-0.11-default -k vmlinuz-3.0.76-0.11-default -f iscsi

g) The previous step did not include the iSCSI script to log into the boot iSCSI target. To include it automatically, boot the system using the Emulex Open-iSCSI driver. Reboot the system and wait until it fails to mount root and exit into a command shell.

h) Run the following command from the command shell to log into the boot target:

```
iscsiadm -m fw -l
```

i) Exit the shell using 'exit' for the system to continue to boot with the iSCSI target.

j) Regenerate the /boot/initrd to include iscsi.sh. iscsi.sh is needed for the initialization script to log into the iSCSI boot target. This is necessary because mkinitrd will include iscsi.sh in the initrd only if there is an entry such as /sys/firmware/iscsi_boot*, which can only be created under Open-iSCSI boot.

k) Reboot the system again. It should boot normally.

l) If the system has other non-boot iSCSI targets, follow the Open-iSCSI instruction to log into those targets.

3. Red Hat does not officially support saving kdump files to iSCSI targets. In RHEL 6.4 and later, to save kdump files to iSCSI targets:

   a) The system must reserve at least 256 MB for the crash dump. Issue the kernel command with the option "crashkernel=256M". Reboot the system for this change to take effect.

   b) Copy the initramfs image to the initrd-kdump image, or build a new initramfs image using Dracut and rename it to initrd-kdump.

   c) Restart the kdump service.

4. The Emulex Ethernet driver for OneConnect adapters supports updating the firmware image in the UCNA flash through the request_firmware interface in Linux. For Oracle Linux 5 Unbreakable Enterprise Kernel systems, an updated version of the ethtool package (ethtool-6-4.0.1.el5.<arch>.rpm) is required. If this package is not already installed on your system, it is available from the Oracle Linux 5 LATEST channel on the Unbreakable Linux Network (Advisory ELBA-2013-2544 released Aug. 29, 2013) and from Oracle's public yum at http://public-yum.oracle.com/repo/OracleLinux/OL5/latest.

   Note: Package updates are not required in Oracle Linux 6 Unbreakable Enterprise Kernel systems.

   You can perform this update when the UCNA is online and passing network or storage traffic.

   To update the Ethernet firmware image:

a) Copy the latest firmware image to the /lib/firmware directory using the following command:

```
# cp be3flash.ufi /lib/firmware
```

b) Start the update process using the following command:

```
# ethtool -f eth<X> be3flash.ufi 0
```

c) Reboot the system to enable the new firmware image.

Note: References to OCe11100 series products also apply to OCe11100R series products.