

# newbus flaw

**Source:** <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-05/0021.html>

---

**From:** Dag-Erling Smørgrav (*des\_at\_des.no*)

**Date:** 05/11/04

To: arch@freebsd.org

Date: Tue, 11 May 2004 16:39:40 +0200

I've found what I believe is a serious flaw in newbus.

When a driver that has a `DEVICE_IDENTIFY` method is loaded, the identify method is called. If it finds supported hardware, it uses `BUS_ADD_CHILD` to notify the parent bus of the presence of that hardware. At some later point, during a bus rescan, the attach routine is called for each device that was identified in this manner.

When the driver is unloaded, the device is detached, but it remains on the bus's list of child devices. The next time the module is loaded, its `DEVICE_IDENTIFY` method is called again, and incorrectly adds a second child device to the bus, because it does not know that one already exists.

There is no way for `DEVICE_IDENTIFY` to check if a matching child already exists on the bus, or for the module's event handler to unlist the child when unloading.

The first time you load the module, you get `foo0`; the second time, you get `foo0` \*and\* `foo1` referencing the same physical device; the third time, you get `foo0`, `foo1`, and `foo2`, etc.

I've also seen something similar happen when multiple `ndis` drivers are loaded; the first one re-attaches to the hardware when the second one is loaded.

DES

--

Dag-Erling Smørgrav - des@des.no

---

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"