wl-kmod

Source RPMs for Broadcom drivers

Please note: The current Broadcom closed-source Linux drivers when built according to the instructions provided here have provided functional wireless capability to some users, but have failed for other users.

Broadcom Corp. provides closed-source IEEE 802.11a/b/g/n Linux drivers for use with Broadcom's BCM4311, BCM4312, BCM4313, BCM4321, BCM4322, BCM4331, BCM4352, BCM4360, BCM43142, BCM43224, BCM43225, BCM43227 and BCM43228 wireless chipsets. We (ELRepo) cannot create and redistribute binary RPMS for these drivers due to the Broadcom licensing restrictions (i.e., liability issues). However, we do make available no-source SRPMS (.nosrc.rpm) which contain the framework required to build the binaries, but lack the proprietary code and are not subject to Broadcom's licensing restrictions.

These no-source SRPMS can be used by the end user to build kABI-compatible binary drivers for local use. These kABI-compatible binaries should be usable after future kernel updates, so there is no need to recompile them for each new kernel. Please note that the nosrc.rpm package does not build kmod-wl for kernel-xen and kmod-wl cannot be built on a system running kernel-xen.

To build a kmod-wl binary, you will need to (1) configure a build tree, (2) download the wl-kmod*nosrc.rpm for your EL release, (3) download the closed-source tarball from Broadcom for your architecture (i.e., 32-bit vs 64-bit) into the build tree and (4) initiate the kmod-wl build. For convenience, the kernel development package for the running kernel will be used in the build process, as opposed to our normal policy of building against an older kernel development package.

The kmod-wl package can be built on any system running the same EL release/architecture as the target system. If you are building on the target system and lack network connectivity, then the downloads can be made onto a USB flash drive and copied onto your system.

Below are the detailed instructions on how to build the binary packages from our no-source SRPM. If you are already familiar with using rpmbuild to build packages, then this process should only take you 5-10 minutes to complete.

Kmod-wl Usability Across EL Point Releases (or "Why doesn't my wireless work after the last point release

update?″):

If you are only interested in building kmod-wl for the latest kernel, then you do not really need to read this section, but it would not hurt to understand the underlying issues.

The EL6 and EL7 kmod-wl binaries are typically only functional within a range of EL point releases (e.g., 6.8-6.10, 7.5-7.6) due to changes in the wireless stack-related kernel API of the EL 6 and EL 7 kernels. RHEL backports components of recent kernels so that more modern kernel features and security fixes are available to the user. These backports become most notable when a new point release becomes available.

Our kmods continue to work after updating the kernel as long as the Kernel Application Binary Interface (kABI) remains unchanged with respect to our packages. EL point releases sometime break this compatibility. For the kmod-wl package, we make changes to the wl-kmod srpm to accommodate these changes. One down side is that the newly built kmod-wl only works within the newest RHEL point release after a wl-kmod srpm update. To maintain backward compatibility of the wl-kmod srpm, we craft it to build slightly different kmod-wl packages depending on the point release kernel used to build it.

If your interest is to only use the newest kernel, then you can just boot to the newest kernel and follow the instructions (below) to build a usable kmod-wl. However, if you need to continue using an older kernel, then you can use the updated wl-kmod srpm to build a kmod-wl package that works with the older kernel.

How to do that: If you build kmod-wl from the updated srpm after booting the latest kernel, then you can save that kmod-wl for use with the latest kernel. Likewise, you can build kmod-wl from the updated srpm after booting an older point release kernel and then save that kmod-wl for use with the older point release kernel. (For users familiar with using rpmbuild, you can also use the " --define 'kversion TARGET-KVERSION ' " option to avoid booting different kernels.) Unfortunately, you can only have one of those kmod-wl packages installed at a time, so save both kmod-wl packages, but only install one of them. Note that they have exactly the same version (i.e., their rpm names are identical), so save them in separate folders.

For example, you can build kmod-wl under an EL 7.2 kernel and use it with EL 7.2 kernels, but it will not function when you boot EL 7.1 or EL 7.3 kernels. Likewise, you can build kmod-wl under an EL 7.3 kernel and use it with EL 7.3 kernels, but it will not function when you boot EL 7.1 or EL 7.2 kernels.

If you need a kmod-wl package that works within one of the following ranges, then boot a kernel in that range and build/install kmod-wl as described below.

Supported EL 6 Point Release Ranges: (1) 6.3 - 6.4, (2) 6.5 - 6.6, (3) 6.7, (4) 6.8 - 6.10

Supported EL 7 Point Release Ranges: (1) 7.0, (2) 7.1, (3) 7.2, (4) 7.3, (5) 7.4, (6) 7.5 - 7.6

Build and install kmod-wl for EL6/7

Please note:

If your current kmod-wl package works with an older kernel and wireless networking is your only Internet access method,
then boot the older kernel and download the new srpm and Broadcom source before proceeding.

1) Install needed tools/packages:

1a) EL6:

```
# yum groupinstall 'Development Tools'
```

```
# yum install redhat-lsb kernel-abi-whitelists
```

```
# yum install kernel-devel-$(uname -r)
```

EL 6.4 - 6.6 Note: Due to a bug in redhat-rpm-config-9.0.3-42.el6, a modification is required to fix the
kabi-whitelist target in /usr/lib/rpm/redhat/find-requires.ksyms before you build kmod-wl.

Please run as root (as a single copy/paste):

```
[[ $(rpm -q redhat-rpm-config|grep 9.0.3-42|wc -l) -gt 0 ]] && sed -i \
's@/lib/modules/kabi/kabi_whitelist@/lib/modules/kabi-current/kabi_whitelist@g' \
/usr/lib/rpm/redhat/find-requires.ksyms
```

The above modification is not needed for EL 6.7 or later.

1b) EL7:

```
# yum group install 'Development Tools'

# yum install redhat-lsb kernel-abi-whitelists

# yum install kernel-devel-$(uname -r)
```

EL 7.0 Note: Due to a bug in redhat-rpm-config-9.1.0-63.el7, a modification is required to fix the
kabi-whitelist target in /usr/lib/rpm/redhat/find-requires.ksyms before you build kmod-wl.

Please run as root (as a single copy/paste):

```
[[ $(rpm -q redhat-rpm-config|grep 9.1.0-63|wc -l) -gt 0 ]] && sed -i \
's@/lib/modules/kabi/kabi_whitelist@/lib/modules/kabi-rhel70/kabi_whitelist@g' \
/usr/lib/rpm/redhat/find-requires.ksyms
```

The above modification is not needed for EL 7.1 or later.

2) As a regular user (not as root), configure a build tree and minimal .rpmmacros:

```
$ mkdir -p ~/rpmbuild/{BUILD,RPMS,SPECS,SOURCES,SRPMS}

$ echo -e "%_topdir $(echo $HOME)/rpmbuild\n%dist .el$(lsb_release -s -r|cut -d"." -f1).local" >> ~/.rpmmacros
```

3) Download wl-kmod*nosrc.rpm

For EL6: http://elrepo.org/linux/elrepo/el6/SRPMS/wl-kmod-6_30_223_271-3.el6.elrepo.nosrc.rpm

For EL7: http://elrepo.org/linux/elrepo/el7/SRPMS/wl-kmod-6_30_223_271-5.el7.elrepo.nosrc.rpm

4) Download the Broadcom driver matching your architecture (i.e., 32-bit vs 64-bit):

from: http://www.broadcom.com/support/802.11 (scroll down to "Linux® STA 32-bit (or 64-bit) drivers")

to: ~/rpmbuild/SOURCES/

5) Build kmod-wl as a regular user (not as root):

```
$ rpmbuild --rebuild --define 'packager <your-name>' /<path-to-nosrc.rpm>/wl-kmod*nosrc.rpm
...
Wrote: /home/<user>/rpmbuild/RPMS/x86_64/kmod-wl-<version>.x86_64.rpm
...
+ exit 0
```

6) If ndiswrapper is installed and is no longer needed, then remove it:

```
# yum remove \*ndiswrapper\*
```

7) Install kmod-wl:

Note for EL7: Recent hardware is often shipped with an Unified Extensible Firmware Interface (UEFI) in which Secure Boot is enabled. The self-compiled kmod-wl is NOT signed for Secure Boot and the corresponding wl module will not load on an UEFI Secure Boot-enabled system. You may see an error message to the effect that "Required key not available". If you wish to continue using Secure Boot in combination with EL7/kmod-wl, then you will need to sign the kmod-wl.rpm you have created using your own private Secure Boot key and install the corresponding public key on your system as explained in Signing Kernel Modules for Secure Boot. Otherwise, you can disable Secure Boot in the UEFI configuration to avoid this issue.

```
# rpm -Uvh /path-to-rpm/kmod-wl*rpm
```

OR

```
# yum --nogpgcheck localinstall /path-to-rpm/kmod-wl*rpm
```

Please note: It is recommended, but is not critical, that you uninstall any older version of kmod-wl (i.e., rpm -e kmod-wl) before installing the new kmod-wl package.

8) Reboot or to start wireless now:

8a) EL6:

```
  # modprobe -r b43 b43legacy ssb wl lib80211
  # modprobe -r bcma (Note: needed for EL 6.4 and later)
  # modprobe lib80211_crypt_tkip
  # modprobe wl
```

8b) EL7:

```
  # modprobe wl
```

9) Store kmod-wl*rpm for safe keeping

10) Optional - Remove the build tree:

```
$ rm -rf ~/rpmbuild
```

Page last modified on Friday 04 of January, 2019 09:21:38 MST