Kernel Modules

LPIC-2: Linux Engineer (201-450)

Objectives:

At the end of this episode, I will be able to:

1. Describe Linux kernel modules and their intended purpose.
2. Utilize lsmod, modinfo, insmod, rmmod, and modprobe to install, monitor, and remove modules.
3. Describe how udev assigns device names and override names if necessary.

> Additional resources used during the episode can be obtained using the download link on the overview episode.

- Kernel Modules
  - The Linux kernel is monolithic
  - You do not want to have every hardware driver loaded in there
  - Modules break things up
  - Modules can be loaded/unloaded as needed
- Viewing loaded modules
  - `lsmod`
  - `modinfo <module_name>`
  - `modinfo e1000`
  - `/lib/modules`
- Loading and unloading modules
  - `insmod`
    - Temporarily enables a module
    - `sudo insmod /lib/modules/5.8.0-45-generic/kernel/drivers/net/ethernet/amazon/ena/ena.ko`
    - `lsmod | grep ena`
  - `rmmod`
    - Removes a module
    - `sudo rmmod ena`
    - `lsmod | grep ena`
  - `modprobe`
    - Loads and unloads modules
    - Provides dependency tracking
    - `sudo modprobe -a ena`
    - `sudo modprobe -r ena`
- Verifying modules
  - Examine the logs
    - `dmesg`
  - List detected hardware
    - `lsusb`
    - `lspci`
    - `lsdev`

        - `sudo apt install procinfo`

- *udev*
    - Assigns device names
    - Follows pre-defined rules in the kernel
    - Can be overridden
        - `/etc/udev/rules.d`
    - Example: Rename ens33 to eth0
        - `sudoedit /etc/udev/rules.d/70-persistent-ipoib.rules`
        - `SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:dd:2e:8e", ATTR{type}=="1", KERNEL=="ens33", NAME="eth0"`
        - `sudo udevadm control --reload-rules && udevadm trigger`
    - Monitor changes
        - `udevadm monitor`