NAME

boltd − thunderbolt device managing system daemon

SYNOPSIS

boltd [*OPTIONS*]

DESCRIPTION

boltd is the thunderbolt device manager daemon. Its goal is to enable the secure and convenient use of thunderbolt devices by using the security features of modern thunderbolt controllers. It provides the org.freedesktop.bolt name on the system bus. boltd is autostarted via systemd/udev if a thunderbolt device is connected.

The thunderbolt I/O technology works by bridging PCIe between the controllers on each end of the connection, which in turn means that devices connected via Thunderbolt are ultimately connected via PCIe. Therefore thunderbolt can achieve very high connection speeds, fast enough to even drive external graphics cards. The downside is that it also makes certain attacks possible. To mitigate these security problems, the latest version — known as Thunderbolt 3 — supports different **security levels**:

none
: No security. The behavior is identical to previous Thunderbolt versions.

dponly
: No PCIe tunnels are created at all, but DisplayPort tunnels are allowed and will work.

user
: Connected devices must be authorized by the user. Only then will the PCIe tunnels be activated.

secure
: Basically the same as user mode, but additionally a key will be written to the device the first time the device is connected. This key will then be used to verify the identity of the connected device.

usbonly
: One PCIe tunnel is created to a usb controller in a thunderbolt dock; no other downstream PCIe tunnels are authorized (needs 4.17 kernel and recent hardware).

The primary task of **boltd** is to authorize thunderbolt peripherals if the security level is either user or secure. It provides a D−Bus API to list devices, enroll them (authorize and store them in the local database) and forget them again (remove previously enrolled devices). It also emits signals if new devices are connected (or removed). During enrollment devices can be set to be automatically authorized as soon as they are connected. A command line tool, called boltctl(1), can be used to control the daemon and perform all the above mentioned tasks.

The pre−boot access control list (**BootACL**) feature is active when supported by the firmware and when *boltd* is running on a new enough Linux kernel (>= 4.17). The *BootACL* is a list of UUIDs, that can be written to the thunderbolt controller. If enabled in the BIOS, all devices in that list will be authorized by the firmware during pre−boot, which means these devices can be used in the BIOS setup and also during Linux early boot. NB: **no device verification** is done, even when the security level is set to *secure* mode in the BIOS, i.e. the maximal effective security level for devices in the *BootACL* is only *user*. If *BootACL* support is present, all new devices will be automatically added. Devices that are *forgotten* (removed from *boltd*) will also be removed from the *BootACL*. When a controller is offline, changes to the *BootACL* will be written to a journal and synchronized back when the controller is online again.

IOMMU support: if the hardware and firmware support using the input–output memory management unit (IOMMU) to restrict direct memory access to certain safe regions, boltd will detect that feature and change its behavior: As long as iommu support is active, as indicated by the iommu_dma_protection sysfs attribute of the domain controller, new devices will be automatically enrolled with the *iommu* policy and existing devices with *iommu* (or *auto*) policy will be automatically authorized by boltd without any user interaction. When iommu is not active, devices that were enrolled with the *iommu* policy will not be authorized automatically. The status of iommu support can be inspected by using **boltctl domains**.

OPTIONS

−h, −−help
Prints a short help text and exits.

−−version
Shows the version number and exits.

−r, −−replace
Replace the currently running boltd instance.

−−journal
Force logging to the journal.

−v, −−verbose
Print debug output.

ENVIRONMENT

RUNTIME_DIRECTORY
Specifies the path where the daemon stores data that only has to live as long as the current boot. Will be set automatically when started via systemd (>= 240). If not set the default path for runtime data is */run/boltd*.

STATE_DIRECTORY
Specifies the path where the daemon stores device information, including the keys used for authorization. Overwrites the path that was set at compile time. Will be set automatically when started via systemd (>= 240).

BOLT_DBPATH
Same as STATE_DIRECTORY but takes precedence over that, if set.

EXIT STATUS

On success 0 is returned, a non−zero failure code otherwise.

AUTHOR

Written by Christian Kellner <ckellner@redhat.com>.

SEE ALSO

boltctl(1)