Thunderbolt 3 and GNU/Linux

FOSDEM 2018

Christian Kellner, PhD
Desktop Hardware Enablement
04/02/2018
What is this THUNDERBOLT™, anyway?
“The USB-C that does it all”

* Intel*

* https://thunderbolttechnology.net/
Thunderbolt 3 — Overview

- USB type C connector (one port to confuse them all)
- 40 Gb/s
- 4 PCI Express (Gen 3) lanes
- 8 DisplayPort (1.2) lanes
- Native USB 3.1
- Daisy-chain up to 6 devices
- Up to 100 W for charging, 15W for devices
- Networking, external Graphic
- Docks, docks, docks
Thunderbolt 3 — Connection Modes

**USB ONLY**
Active when USB devices are plugged in. Behaves as a normal USB-C 3.1 port.

**DISPLAYPORT ONLY**
Switch pins of USB-C into DP alternate mode. TB will act as a router for DP data from GFX to USB-C port.

**DP & USB MULTI-FUNCTION**
One high-speed pair is used for DP. The other high-speed pair is used for USB 3.1.

**THUNDERBOLT 3**
All 4 high-speed links active (at 10/20 Gbps). max 4 PCIe Gen 3 lanes max 2 DisplayPort links.

**POWER DELIVERY & CHARGING**

**THUNDERBOLT NETWORKING**
Thunderbolt — Security ???

Thunderbolt is PCIe → DMA→ DMA attacks

https://github.com/ufrisk/pcileeuch
Thunderbolt 3 — Security Modes

NONE
All devices are authorized by default.

USER
Thunderbolt devices need to authorized. Only then are PCIe lanes activated.

DP ONLY
Display Port only. You guessed right.

SECURE
Thunderbolt devices need to authorized. Their identity can be verified via a key.
Thunderbolt 3 — Security Modes

In the land of the dialogs … … no we are not doing that.

New Thunderbolt™ devices have been attached

Thunderbolt™ devices were attached. In order to approve them for use this application must be run with administrator privileges. Click OK to run the Thunderbolt™ software with administrator privileges.

Approve Thunderbolt™ Devices

The following Thunderbolt™ device chain has been plugged in and one or more devices require your permission to connect to this system.

Select the devices you wish to connect:

Plugable, TBT3-DP2X

Note: Selecting "Do Not Connect" will prevent that device in the chain from being used on the system. Install the driver included with the device before approving the device.

Always Connect

OK
Thunderbolt and GNU/Linux
Thunderbolt & GNU/Linux

Overview

- **sysfs/udev**
- **boltd** System daemon
- **D-Bus**
- **Linux 4.13**
- **boltctl**
- **other DE integration**
- **gnome-control-center**
- **gnome-shell**
- **boltctl**
Kernel Interface
Linux kernel 4.13+ provide a sysfs interface

```
/sys/bus/thunderbolt/
 └── devices
     ├── domain0 → 0-0/ security subsystem@ uevent […]
     │      └── 0-0 → 0-1/ authorized device device_name vendor_name unique_id […]
     │      └── 0-1 → 0-301/ authorized […] key […] unique_id
     └── 0-301 → […] nvm_active2/ nvm_non_active2/ nvm_version nvm_authenticate

# echo 1 > /sys/bus/thunderbolt/devices/0-1/authorized

# key=$(openssl rand -hex 32)
# echo $key > /sys/bus/thunderbolt/devices/0-1/key
# echo 1 > /sys/bus/thunderbolt/devices/0-1/authorized

# echo $key > /sys/bus/thunderbolt/devices/0-1/key
# echo 2 > /sys/bus/thunderbolt/devices/0-1/authorized
```
Thunderbolt firmware updates
fwupd & Linux Vendor Firmware Service (LVFS)

# get current version
nvm_version

# write new firmware to
nvm_non_active2/nvmem

# start updating
nvm_authenticate

* https://fwupd.org/
Thunderbolt & GNU/Linux

boltd

- System daemon, activated on demand
- D-Bus API to manage devices, signal device “changes”
- Authorize, enroll (authorize and store)
- Polkit to secure the D-Bus API
- Device “database” of previously enrolled devices and their policy
- Paranoid (now fortify) mode

- Needs a policy agent to do the initial authorization, enrollment
boltd
D-Bus API: manager interface

- **Address:** unix:path=/var/run/dbus/system_bus_socket
- **Name:** org.freedesktop.bolt
- **Unique name:**

**Object path**
/org/freedesktop/bolt

**Interfaces**
- org.freedesktop.DBus.Introspectable
- org.freedesktop.DBus.Peer
- org.freedesktop.DBus.Properties
- org.freedesktop.bolt1.Manager

**Methods**
- `DeviceByUid` (String uid) ➔ (Object Path device)
- `EnrollDevice` (String uid, UInt32 policy, UInt32 flags) ➔ (Object Path device)
- `ForgetDevice` (String uid) ➔ ()
- `ListDevices` () ➔ (Array of [Object Path] devices)

**Properties**
- `Boolean Fortify` (read / write)
- `Boolean Probing` (read)
- `UInt32 DefaultPolicy` (read)
- `UInt32 Version` (read)

**Signals**
- `DeviceAdded` (Object Path)
- `DeviceRemoved` (Object Path)
boltd
D-Bus API: manager interface

```
Interfaces

- org.freedesktop.DBus.Introspectable
- org.freedesktop.DBus.Peer
- org.freedesktop.DBus.Properties
- org.freedesktop.bolt1.Device

Methods

- Authorize (UInt32 flags) ➞ ()

Properties

- Boolean Stored (read)
- String Name (read)
- String Parent (read)
- String SysfsPath (read)
- String Uid (read)
- String Vendor (read)
- UInt32 Key (read)
- UInt32 Policy (read)
- UInt32 Security (read)
- UInt32 Status (read)
- UInt32 Type (read)
```
**boltctl**

**cli interface**
gnome-shell
Acts as a policy agent

Listen to “device-added” D-Bus signal from bolt
d

user logged in & session unlocked

yes

user is admin

yes

Enroll device

no

Notification: new Unauthorized device

Polkit

admin authorization
gnome-shell
Acts as a policy agent
gnome-shell
provide UI feedback about thunderbolt bus activity
gnome-control-center
manage devices, provide feedback
**gnome-control-center**

manage devices, provide feedback

---

Authorization issues
One or more thunderbolt devices are connected but not authorized and therefore will not work properly. [Learn more]

Thunderbolt devices
- **Lenovo ThinkPad Thunderbolt 3 Dock**
  - The device needs authorization. Please re-plug the device.

---

This device is connected but not authorized.
Re-plug the device to authorize it. [Learn more]

- **Lenovo ThinkPad Thunderbolt 3 Dock**
  - Status: Connected
  - Stored: No
  - Details:
THANK YOU

github.com/gicmo/bolt

christian.kellner.me