Firmware updates for pfSense and OPNsense with fwupd/LVFS

FOSDEM22

Norbert Kamiński

3MDEB

Agenda

- Project genesis
- Overall information about fwupd
- fwupd tool architecture
- pfSense proof of concept
- OPNSense proof of concept
- Existing issues and future steps
- Q&A

Many thanks!

Many thanks to my colleague - Michał Kopeć, who did OPNsense proof of concept work.

$ whoami

Norbert Kamiński
Embedded Systems Engineer

- open-source contributor:
  - fwupd
  - meta-pcengines
- scope of interests:
  - firmware upgrade tools
  - virtualization
  - embedded Linux

- norbert.kaminski@3mdeb.com
- linkedin.com/in/norbert-kami%C5%84ski/
- facebook.com/nkaminski3
- @asiderr

Who we are?


OpenPOWER™

- coreboot licensed service providers since 2016 and leadership participants
- UEFI Adopters since 2018
- Yocto Participants and Embedded Linux experts since 2019
- Official consultants for Linux Foundation fwupd/LVFS project since 2020
- IBM OpenPOWER Foundation members since 2020

Project genesis

- Routers always were important target for various adversaries, therefore they must be appropriately protected
- It is almost impossible to update device firmware without proper tools
- Our products - Basic firmware update integration and open-source firmware Dasharo will benefit from this integration

DASHARO

fwupd - overall information

- fwupd project can query supported hardware for the current firmware versions and also deploy new firmware versions to devices
- LVFS is a secure web service that provides information about available firmware updates. It can be used by the OEM's to upload firmware archives downloaded by the users
- fwupd could be a good way to simplify the deployment of firmware updates for firewall device and to facilitate the update process for end-users

fwupd/LVFS architecture

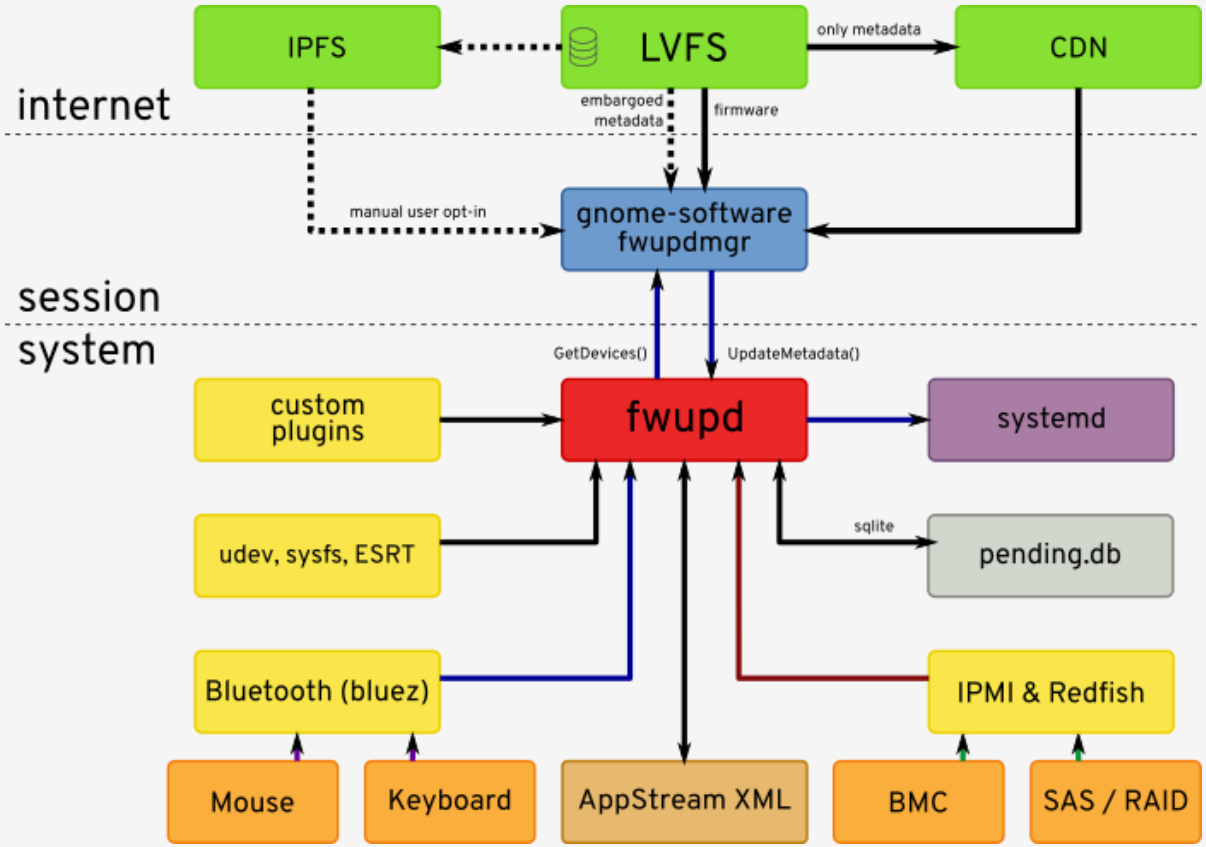
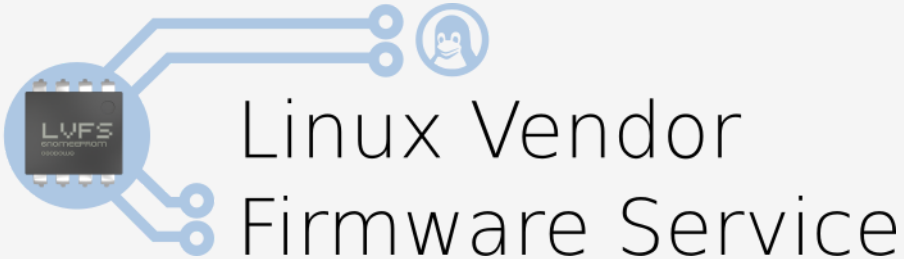


Image source: https://lvfs.readthedocs.io/en/latest/intro.html

Linux Vendor Firmware Service (LVFS)



- The LVFS is a secure web service that is used by OEM's to provide firmware updates
- The LVFS provides metadata that contains information about possible updates
- The firmware updates are packed into cabinet archives. The archive contains the firmware blob, information about the update, and `jcat` file, which is used to verify the firmware updates
- A manufacturer is signing the firmware and this sign is verified during the update

pfSense proof of concept

pfSense is a free and open-source firewall and router that also features unified threat management, load balancing, multi-WAN, and more. It is based on FreeBSD.

Hardware and software stack:

- PC Engines apu2 - Router/firewall board
- pfSense 2.5.2
- fwupd 1.7.4
- FreeBSD 12.2 VM

pfSense proof of concept

- The first step was the installation of the pfSense 2.5.2 at apu2
- The next step was the enablement of the FreeBSD pkg repo in the pfSense

```
vi /usr/local/etc/pkg/repos/pfSense.conf
```

```
FreeBSD: {
url: "pkg+<http://pkg.freebsd.org/FreeBSD:12:amd64/release_2>",
signature_type: "fingerprints",
fingerprints: "/usr/share/keys/pkg",
mirror_type: "srv",
enabled: yes
}
```

- As it is mentioned in the documentation, the packages for pfSense should be compiled in the FreeBSD VM

pfSense proof of concept

- We used FreeBSD 12.2 virtual machine to create fwupd package for pfSense 2.5.2
- We created the fwupd package and its dependencies in [3mdeb's FreeBSD ports fork]

```
PORTNAME=    fwupd
DISTVERSION=    1.7.5
GH_TAGNAME=    38bab8f
CATEGORIES=    sysutils

MAINTAINER=    norbert.kaminski@3mdeb.com
COMMENT=    Update firmware automatically, safely, and reliably

LICENSE=    LGPL21

BUILD_DEPENDS=    gtkdoc-scan:textproc/gtk-doc \
  help2man:misc/help2man \
  vala:lang/vala \
  ${LOCALBASE}/libexec/fwupd/efi/fwupdx64.efi:sysutils/fwupd-efi \
  ${PYTHON_PKGNAMEPREFIX}gobject3>0:devel/py-gobject3@${PY_FLAVOR}
LIB_DEPENDS=    libcurl.so:ftp/curl \
  libgcab-1.0.so:archivers/gcab \
  libgnutls.so:security/gnutls \
[...]
USES=        gnome libarchive meson pkgconfig python:3.8+ shebangfix sqlite
USE_GITHUB=    yes
[...]
MESON_ARGS=    -Dgudev=false \
      -Dplugin_amt=false \
      -Dplugin_dell=false \
      -Dplugin_emmc=false \
      -Dplugin_nvme=false \
      -Dplugin_parade_lspcon=false \
[...]
.include <bsd.port.mk>
```

pfSense proof of concept

- We installed fwupd dependencies in the FreeBSD VM and in the pfSense to comply with all of the build and runtime dependencies

```
pkg install -y git python3 glib meson pkgconf gobject-introspection \
          vala gtk-doc json-glib gpgme gnutls sqlite3 curl gcab libarchive \
          libgpg-error gettext-tools gtk-update-icon-cache atk pango \
          binutils gcc protobuf-c efivar
```

- The next step was cloning our fork in the FreeBSD VM:

```
git clone <https://github.com/3mdeb/freebsd-ports.git> -b fwupd-pfsense --depth 1 /usr/ports
```

- Then we built fwupd dependencies and fwupd package that were not currently in the FreeBSD ports - `devel/libgusb`, `textproc/libjcat`, `textproc/libxmlb`, and `sysutils/fwupd`

pfSense proof of concept

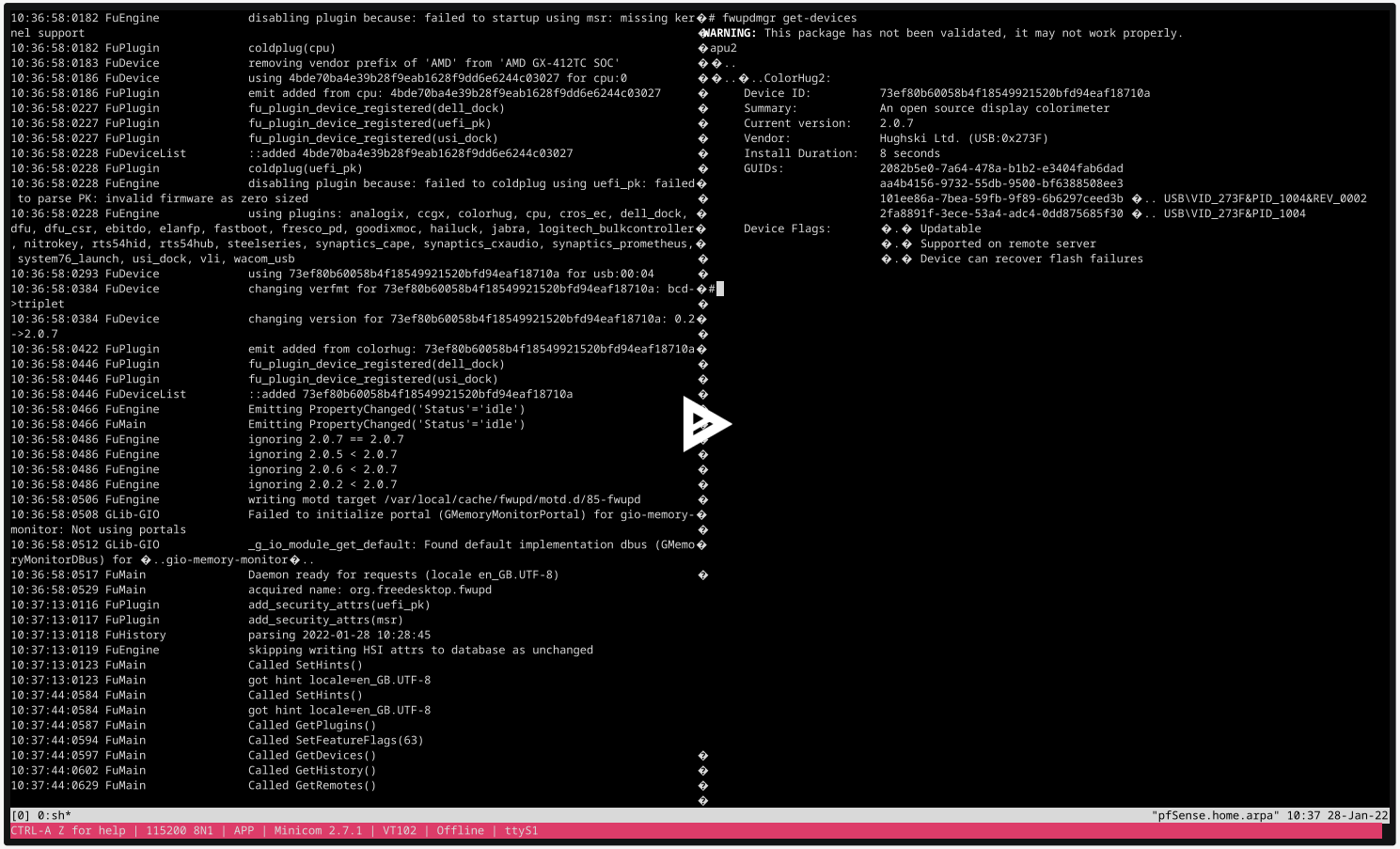
- The created packages were sent to the pfSense via scp
- Then we installed the fwupd dependencies and fwupd itself:

```
pkg add ./libgusb-0.3.7.pkg ./libjcat-0.1.8.pkg ./libxmlb-0.3.6.pkg ./fwupd-1.7.5.pkg
```

- The last step was running `dbus`:

```
service dbus onestart
```

pfSense proof of concept - Demo



OPNsense proof of concept

OPNsense is an open-source, FreeBSD-based firewall and routing software developed by Deciso. Our proof of concept work was based on OPNsense 21.1, which is based on FreeBSD 12.1 with HardenedBSD patches

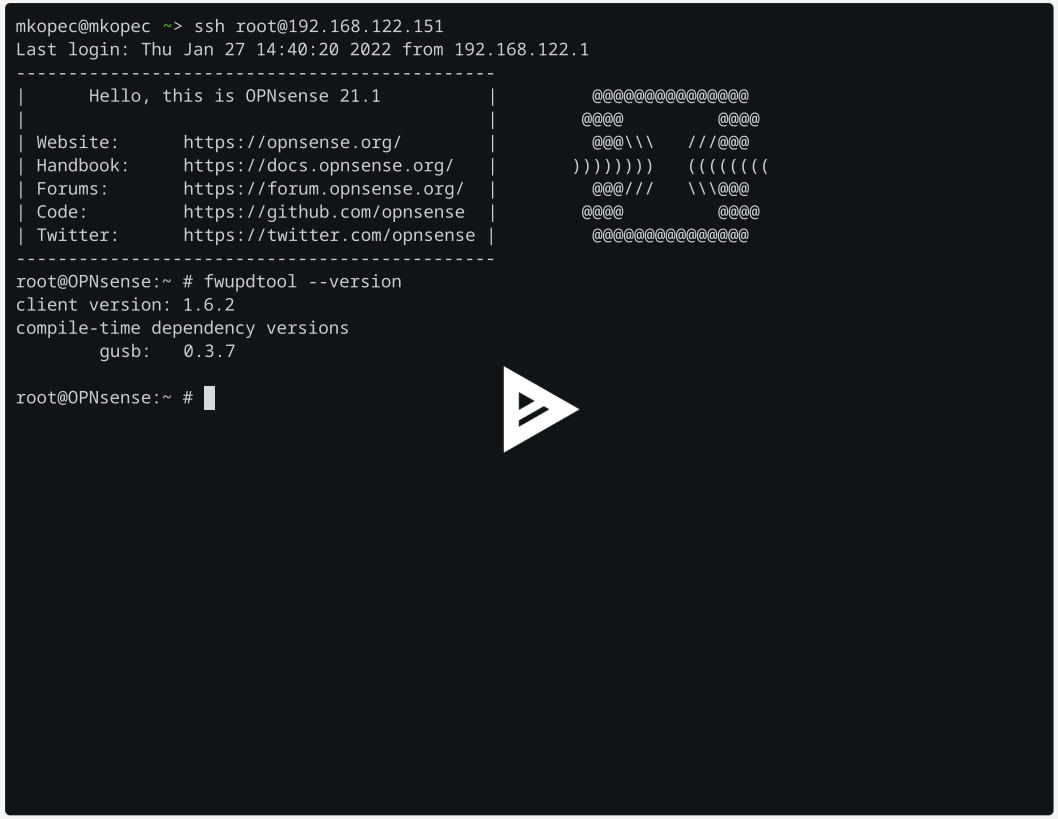
Hardware and software stack:

- PC Engines apu2 - Router/firewall board
- OPNsense 21.1
- fwupd 1.6.2

OPNsense proof of concept

- Due to the fact that OPNsense and pfSense are FreeBSD based software, most of the pfSense PoC work could be directly moved to OPNsense
- OPNsense uses pkg as the package manager and supports the usual FreeBSD build facilities. The official repositories only contain a subset of upstream FreeBSD repositories
- Applying our fwupd port on top of the OPNsense ports tree works mostly without conflicts. It's possible to compile and run fwupd, with some plugins disabled

OPNsense proof of concept



Existing and past issues

Some issues needed solving before fwupd could compile and run:

- Running fwupdtool fails because the file `/etc/os-release` is missing. Creating one manually fixes the issue and lets fwupdtool run.
- On FreeBSD 12.1, including `<malloc.h>` causes the compiler to throw an error. Any instance of `<malloc.h>` needs to be replaced with `<stdlib.h>`. This was fixed in FreeBSD 13.0: https://github.com/freebsd/freebsd-src/commit/bbbed78aaa9ad2d55ed30c15707e7efce92b80d1
- Some parts are missing to compile fwupd with the UEFI support:
  - fwupd-efi needs compiling with gcc10, which isn't available in OPNsense repositories (compiling from ports is required)
  - ESRT support
  - efivar support

Future steps

- Finding funding for the future work
- Fixing known issues
- Adding support for new fwupd plugins (UEFI capsule, flashrom plugin)
- Constant upstream of our work

Thank you for your attention!