Building operating systems optimized for containers, from IoT to desktops and servers

Timothée Ravier
CoreOS engineer at Red Hat

Pass the SALT 2022

Usual caveats and warnings

- This talk is about **community projects** (i.e. not a product)
- Thus reflects **my opinions**
- But I believe the goals are **shared**
- I'm a **contributor** to some of the projects mentioned
    - and some of the underlying technologies used

Breaking News: Software has bugs (!)

- **Memory safety** issues, logic bugs
- **Linux kernel** vulnerabilities
- CVEs & **non** CVE fixes
- etc.

- Can’t **find** them all, can’t **fix** them all
- Can’t “just” **rewrite** everything in <good language>

Well-known workarounds

- **Update** vulnerable software
- Focus on **bug classes** instead of single bugs
- Progressively introduce **better languages** in codebases
- **Defense in depth:**
  - Use **just** what you need
  - **Split** privileges
  - Put as much as possible into a **sandbox**

Goal: Make workarounds usable

- Most users only use the **default** configuration
- Make the default behavior the **secure** option
  - No “secure configuration”
  - No “security focused” distribution
- Make updates a **non-event** and **enabled** by default
- Use a sandbox for **applications** by default

How can we do this?

Limits of package centered systems

- Securing classic **package based** distributions is hard
- Requires **expert knowledge** and time to set up
- Can not provide **at the same time**:
    - lots of packages
    - secure by default packages
- Must select a **smaller** default set
    - Part of **attack surface reduction**

Moving to image based distributions

- Provide a **curated set** of packages by default
- Every system is **the same** for a given version
- Makes **testing** and **reproducing** issues easier
- Updates are **atomic**
- But what do we **create** those images from?

Taking a look at the Fedora Project

- Provides a **stable** and **up to date** software stack
- New release approximately every **six months**:
  - Mostly **security fixes** for stable releases
  - Major **new features** go into the next release
- **Upstream first** for patches and configuration changes

ostree & rpm-ostree: Bridging the gap

- Hybrid image/package system with **atomic upgrades**
- Kind of like **Git** for your operating system
- Create “images” from **existing** packages
- Client side **package layering** and overrides:
  - Add, remove or replace packages **locally**
- Operations are **atomic**, **safe** and **easy to rollback**

OS versioning and filesystem layout

- A **single identifier** for a given system version
  - Example: 36.20220605.3.0
- Uses **read-only** filesystem mounts:
  - Prevents accidents, basic attacks and **real vulnerabilities**
- **Clear distinction** between:
  - **`/usr`** □ distribution content (from packages)
  - **`/etc`** □ system configuration (defaults from packages)
  - **`/var`** □ all local system and user content

Where is this happening?

rpm-ostree based Fedora variants

fedora SILVERBLUE

fedora COREOS

fedora IoT

fedora KINOITE

- Each variant is focused on a specific **use case**
- **Varying** degree of progress toward the stated goals

Common ground for all variants

- Built **100%** from Fedora RPM packages
- System managed by **rpm-ostree**
- Most applications are run in **containers**:
  - **Podman** is included by default
- Enables **decoupling** applications and system updates

Containers & security by default

- Confinement with **SELinux**:
  - **Confined** system services (targeted policy)
  - **Isolation** between containers and container ⇔ host
  - Already **blocked** several real vulnerabilities in runc:
    - CVE-2019-5736: Latest container exploit (runc) can be blocked by SELinux
    - CVE-2021-30465: Mitigated by Default in OpenShift

What is Fedora IoT?

Fedora IoT

- Focused on **IoT** use cases:
  - industrial gateways
  - smart cities
  - analytics with AI/ML
  - a project at home
- Management with **Ansible**

Architectures and devices

- Support for **x86_64**, **aarch64** and **ARMv7**:
  - Only supports devices with **UEFI support**
  - SoCs supported by Fedora (requires **SBBR/EBBR**)
  - ARMv7 support will end with Fedora 37
- Some **example devices** include:
  - NVIDIA Jetson Xavier series
  - Compulabs Fitlet2
  - Solid-run Honeycomb and Hummingboards
  - **Raspberry Pi** series of devices

Security for IoT & Edge devices

- Building on top of **TPM2** devices:
  - **Remote attestation** with Keylime
  - Pin disk encryption to **TPM PCRs** with Clevis
- Auto-updates are configurable:
  - Setup Greenboot to enable **automatic rollbacks**
- **On-boarding** with Zezere:
  - Minimal touch on-boarding for a fleet of devices

Upcoming: Secure on-boarding

- **FIDO Device Onboarding (FDO):**
  - Zero touch **secure provisioning** for IoT
  - Based on **FIDO specification**
  - Easily on-board a large number of devices
- Implemented entirely in **Rust** (client & server):
  - https://github.com/fedora-iot/fido-device-onboard-rs
- Planned for Fedora 37

What is Fedora CoreOS?

Fedora CoreOS

- Successor to two **container-first** OSes:
  - CoreOS Inc’s Container Linux
  - Fedora Atomic Host (from Project Atomic)

- **Incorporates ideas** from both:
  - Provisioning stack & cloud native expertise (CL)
  - Fedora foundation, update stack & SELinux (FAH)

- Focused on **single node** and **clusters** use cases

Philosophy

- **Automatic updates** by default
  - No interaction for administrators

- **Automated provisioning**
  - All nodes start from **same starting point**
  - Use Ignition to provision a node on **first boot**

- **Immutable infrastructure**
  - **Automate** deployment and system configuration
  - Update configs and **re-provision** to apply changes

Platforms and architectures

- Available for a plethora of **cloud/virt platforms**:
  - Alibaba, AWS, Azure, Azure Stack, DigitalOcean, Exoscale, GCP, IBM Cloud, OpenStack, Nutanix, Vultr, VirtualBox, VMware, QEMU/KVM
  - Directly launchable on AWS & GCP
- Several options for **Bare Metal**:
  - Live ISO, PXE (network) boot, 512b/4K native disk images
- Support for **x86_64**, **aarch64** and **s390x**

Reducing the OS footprint

- First step in security hardening: **reducing** attack surface
  - Less software to **track** for security and bug fixes

- Only **essential** system services and administration tools
- Two **container runtimes**: podman & moby-engine
- Only includes Bash: no Python, etc.

Building with safer languages

- Using **memory safe languages** for most of Fedora CoreOS specific additions:
    - **Go**: Butane, Ignition, toolbx, container engines (podman & moby-engine)
    - **Rust**: Afterburn, Zincati, coreos-installer, bootupd, rpm-ostree (in progress), Cincinnati

Fedora CoreOS examples

- **Single node** Matrix server:
  - https://github.com/travier/fedora-coreos-matrix
- **Nomad** cluster:
  - https://github.com/travier/fedora-coreos-nomad
- **Kubernetes** cluster with **OKD**:
  - https://www.okd.io/installation/

What are Fedora Silverblue & Kinoite?

Fedora Silverblue & Fedora Kinoite

- **Desktop** variants with **Wayland** and **Pipewire**
- Fedora **Silverblue**:
  - Featuring the **GNOME** desktop
  - Following the work on Fedora Workstation
- Fedora **Kinoite**:
  - Featuring the **KDE Plasma** desktop
  - Following the work on the KDE Spin

Easy desktop experience for users

- **rpm-ostree** makes system updates a non-event
    - Prepared in the background
    - Applied on reboot
    - Instant rollback when facing issues
- Work in progress in GNOME Software and Plasma Discover to make them **easier to manage**
- End goal is to make them **“transparent”**

More sandboxed applications

- Applications shipped as **Flatpaks**
  - Installation and updates **independent** of system operations

- More and more applications use **Portals**
  - Thus using a **strong sandbox**
  - X.org deprecation will remove the biggest hole in the sandbox

- Major applications **providers**:
  - **Fedora** (FOSS only)
  - **Flathub** (mixed FOSS and proprietary)

Development in and with containers

- Use containers to create **mutable environments** that are independent of the system
- Install any package, development tools, IDEs, etc.
- **Not a security boundary**: a lot is shared with the host

- **toolbx:** Currently Fedora focused but other distributions are planned

- **Distrobox:** Works with most Linux distributions

Future security improvements

Future work: Runtime integrity for ostree

- **rpm-ostree** checks integrity at update time
- Then relies on **filesystem** or **block device** integrity
- Work in progress: **composefs**
    - new “virtual” filesystem
    - modeled around ostree repo format
    - based on **fs-verity**
    - enables **“at access time”** integrity checks

Future work: Boot attestation

- Integrate **Keylime** (in Rust) into other variants (already in IoT):
  - **remote boot attestation** for server use cases
  - **local boot attestation** for desktops (see also tmptotp)

- Improving the user experience with **TPM pinned encryption**:
  - Make Clevis (and Tang) easier for desktops
  - Installer changes and user story for **recovery**

Get involved!

- Fedora IoT: https://getfedora.org/iot/
- Fedora CoreOS: https://getfedora.org/coreos
- Fedora Silverblue: https://silverblue.fedoraproject.org/
- Fedora Kinoite: https://kinoite.fedoraproject.org/

fedora SILVERBLUE

fedora COREOS

fedora IoT

fedora KINOITE