Delta-like Streaming of (encrypted) OTA Updates for RAUC

FOSDEM 2023

Enrico Jörns – e.joerns@pengutronix.de
About Me & Pengutronix

- Embedded software developer
- RAUC co-maintainer
- Integration Team Lead at Pengutronix

Pengutronix

- Embedded Linux consulting & support since 2001
- 7000+ patches in Linux kernel
Structure

- Introduction + RAUC overview
- RAUC bundle format
- HTTP(S) bundle streaming
- Adaptive updates
- Encrypted bundles
- App updates
- Outlook & ecosystem
(OTA) Field Updates
RAUC – Scope

Update bundle generation

Fail-Safe installation on target
RAUC

- Embedded Linux update *framework*
- Fail-safe, authenticated, image-based (A/B) installation

Written in C
(with glib, OpenSSL, curl, ...)

LGPL-2.1 License

https://github.com/rauc/rauc
Bundle Format
Limitations of Initial Bundle Format

- SquashFS containing
  - Manifest, images, hooks, ...
- Attached signature

Bundle Generation
Limitations of Initial Bundle Format

- SquashFS containing
  - Manifest, images, hooks, ...
- Attached signature
- Read entire bundle for verification!
  - Slow, not streamable!
Background – dm-verity

- Linux kernel device mapper
  - Generic block device manipulation abstraction
- Hash-table to authenticate read-only block devices
Verity Bundle Format – Creation

- dm-verity (Merkle) hash tree + root hash generated
- Root hash → trust anchor
  - Stored in manifest
Verity Bundle Format – Creation

- dm-verity (Merkle) hash tree + root hash generated
- Root hash → trust anchor
  - Stored in manifest
- Enveloping signature
  - Contains manifest
Verity Bundle Format – Authentication

- Fast Manifest authentication
  - Authenticates verity root hash!
Verity Bundle Format – Authentication

- Fast Manifest authentication
  - Authenticated random access!
  - Per-chunk verification
  - Authenticated random access!
Bundle Streaming
RAUC Streaming Support

- Full bundle download (by external application)
- Intermediate storage required
- Built-in on-demand download of required bundle data
- No intermediate storage
HTTP(S) Bundle Streaming

- Unprivileged helper process (using libcurl)
- HTTP(S) range requests
  - Supported by all common webservers and many CDNs
HTTP(S) Bundle Streaming

- Unprivileged helper process (using libcurl)
- HTTP(S) range requests
  - Supported by all common webservers and many CDNs
  - Authenticated random access to remote bundle
  - No intermediate storage
Saving Download Bandwidth
Conventional Delta-Updates

- ✓ optimal diff
- ✗ require source + target image
- ✗ allow only step-by-step updates!
RAUC Adaptive Updates – Concept

- Bundle provides optional download optimization
  - Additional data stored in bundle
- RAUC selects appropriate option
  - Downloads only the required data (streaming!)

* always as fallback
Adaptive Update Method – hash-index

- block-hash-index implemented
- file-based variant via rsync + checksums planned
Bundle Encryption
Bundle Encryption

- Hide sensitive data or application IP from third-party

- Two-Stage approach in RAUC
Bundle Payload Encryption – dm-crypt

- We just add another layer...
- Device mapper: dm-crypt (Symmetric with AES-256)
- Transparent decryption during installation
Bundle Payload Encryption

- Block-wise authenticated decryption
- Random access
- ✔ Streamable!
„App”-Updates

(In development)
Application Updates – Situation So Far

- Assumes monolithic system
  - Single (main) application
  - Linked against rootfs libs
- Application: part of rootfs or on separate slot
  - Always update both to have consistent state
‘App’ Updates

- Capability of updating
  - Apps, Containers, Maps, ...
  - Independent of rootfs (w/o reboot)
  - More frequent than OS
  - By different vendors
- Was beyond the scope of RAUC so far..
RAUC Artifact Updates

- New ‘artifacts’ slots
  - File- / Directory-based
  - Can hold multiple artifacts
- Atomic updates (symlinks)
- Redundant for dependency on rootfs
- Streaming support through rsync checksum files

https://github.com/rauc/rauc/issues/969
Outlook & Community
Current & Future Developments

- Switch to Meson (already merged)
- Custom meta-data in manifest
  - Exposed in ‘rauc info’ + D-Bus API
- Fine-grained installation progress
  - Tar extract progress

[meta.pengutronix]
location=Brussels
release-notes=https://...
release-channel=beta

[meta.vendor]
key=value
Ecosystem: rauc-hawkbit-updater

- **Eclipse hawkBit:**
  Open Source back-end framework for software rollouts
- **RAUC adapter in C** started by Prevas (2018)
- **Moved to RAUC GitHub org** (2020)

https://github.com/rauc/rauc-hawkbit-updater
Community: meta-rauc-community

- Bitbake layer collections for example integrations
- Maintained by Leon Anavi
- Supported boards:
  - qemux86-64
  - raspberrypi
  - Sunxi
  - Tegra

https://github.com/rauc/meta-rauc-community
Community: RAUC-related Projects / Products

- Valve Steam Deck
  - RAUC + desync (casync variant in Go)
  - Patches mainlined by Collabora
- Home Assistant Operating System
  - Buildroot updated with RAUC
- Oniro
  - Eclipse project for distributed systems
- DB Linux4ICEs
  - Linux distro for ICE train info panels

https://www.heise.de/select/ct/2022/27/2223808321839973008
Thank You!

Questions?

Join the discussion and get help on: #rauc IRC/Matrix channel
Timeline

Project Start: 2015
Initial Release: 2017
Bootloader Updates: 2020
Verity bundles: v1.5, 2021
HTTP(S) streaming: v1.6, v1.7, v1.8, 2022
Encryption: v1.9, v1.10, 2023
Artifact Updates: today
RAUC – Configuration Basics

Update manifest → in bundle

System configuration → on target

→ Introspectable!