Microsoft Azure

Getting started with ue-rs

- Dongsu Park

- 11. Mar. 2025

Update engine

- Heart of rolling update mechanism of Flatcar
- Consists of 4 parts
  - update_engine_client: command-line tool that sends queries to daemon
  - update_engine: daemon that listens to client requests, relays the requests to a remote Nebraska server
  - delta_generator: command-line tool that generates binary diff (delta) between 2 versions of images. Invoked during build_image step of build scripts.
  - flatcar-postinst: standalone script to take care of postinstall actions in Omaha reponse

Omaha

- Client server communication protocol with XML data body
  - https://github.com/google/omaha/
- Communication between update_engine and Nebraska server
- Data structure defined in protobuf format in update_metadata.proto
  - Translated into actual source files *.pb.{cc,h}, update_metadata.rs

Format of update payload

- Header of every update payload includes a file magic string "CrAU"

- Format of update payload

header	manifest	data blobs	signatures

Ue-rs: rust-reimplementation of update_engine

- Minimal reimplementation of update_engine, which is for historical reasons heavy and complicated.
- Rewriting only essential parts like parsing Omaha protocol from scratch
- Use pure Rust RSA libraries instead of relying on openssl.
- Written in Rust, a huge advantage for memory safety, while update_engine in C++
- Aims to be pluggable for integration with systemd-sysupdate
- Should be integrated with Nebraska

Ue-rs: current status

- Still a proof of concepts
- download_sysext: standalone binary to demonstrate sysext OEM image to parse Omaha response and verify checksum & signatures.
- omaha: library for parsing Omaha messages in dedicated workspace
- update-format-crau: library for verifying RSA signatures in dedicated workspace

Future work (?)

- Fetch, validate, install OS image and extensions
  - Fetch, verify, write partitions, Kernel, sysext images
  - Run postinstall hook
- Full support of Omaha protocol
  - ping, check for updates, generate payloads
- Facilitate machine model
  - Reports states like UPDATE_STATUS_{IDLE,UPDATE_AVAILABLE}
- Implement DBus communication for client-server architecture
  - Based on Rust-native DBus implementation like zbus

Future work for newcomers

- Add missing unit tests
- Resolve dependency update issues (like idna)
- Resolve security issues (like rsa)
- Improve GitHub Actions CI
  - Clippy

Questions?

Thanks!