Secure Partition Runtime Library Status

Ken Liu
Jan 9th
Runtime APIs for Secure Partition

Secure Partition Runtime Library

- PSA FF API
- libc
- RoT Service API
- MISC
Reasons for Creating SPRTL

• The Partitions are isolated – Duplicated Copy of Runtime Library?

• Toolchain provided libc symbols:
  • Some needs to be enhanced: comparing functions.
  • Some needs to be re-implemented: printf and related functions.
  • Auditing?

• Necessary part of the SP build
Secure Partition Runtime – Library Design Guidelines

- Library contains CODE only
- No writeable private data
- Use caller stack
Secure Partition Runtime – What does PSA FF say?

I1. Only Code is executable
I2. Only Private data is writable
I3. Private data must be isolated
I4. CODE and RO should be isolated (optional)
I5. CODE should not be executed by others (optional)
I6. SPM should not execute partition CODE (Optional)
<= We are here:

- Structure is setup with necessary creating of makefiles and linker scripts.
- Part of libc candidates are ready for forwarding – toolchain built-in optimization needs to be avoided before this.
- PSA FF moved to this area for all partitions.
Goal of SP Building

service/
- src/
  > partition.json  Partition Description
  > service.c  Partition Service Logic
- interface/
  > service.h  Headers for Clients
- lib/
  libsprt.a  Secure Partition Runtime Library
- generated/
  - psa_manifest/
    > sid.h  Service ID generated from manifest for Clients
    > pid.h  Partition ID for Partition Interactive
    > signals.h  Generated Signals for Partition internal usage
    > partition_info.h  Partition Information for Partition Database
    > partition.ld  Partition position info for linker
Thank You
Danke
Merci
谢谢
ありがとうございます
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
tודה