Qualcomm

Supporting generic restricted dmabuf heap

Prakash Gupta

Sr. Staff Engineer/Mgr, Qualcomm India

quic_guptap@quicinc.com

Introduction

Why protected DMA-BUF Heaps?

Generic protected heaps

QCOM secure video playback

Introduction - Restricted dmabuf heaps

- DMA-BUFs
  - Very useful for sharing buffers between multiple devices, avoiding copies.[1]
- DMA-BUF Heaps
  - Provides an userland interface to allocate DMA-BUFs that point to specific types of memory [1]
- Secure DMA-BUF Heaps:
  - Should we call - Secure heap/ Restricted heap / Protected heap?
  - The specific types of memory here is memory with some restrictions from HLOS access. Eg: HLOS Read-Only, or No HLOS access allowed

- Methods to restrict buffer access
  - External Protection Unit (XPU) [2]
  - Type -1 hypervisor stage 2 based restriction

    eg: qcom_scm_assign_mem()

[1] https://static.linaro.org/connect/lvc21/presentations/lvc21-120.pdf

[2] https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/an-introduction-to-access-control-on-qualcomm-snapdragon-platforms.pdf

Why protected DMA-BUF Heaps?

Use cases requiring protected buffers can take benefit of dmabuf using protected dmabuf heaps:

Below are few examples:

- Secure Video playback
- FastRPC CPZ (Compute Protection Zone) - Video post-processing

These can also be achieved using custom userland/kernel driver interface, there are few challenges.

Why protected dma-buf heaps?

Stop gap example in absence of restricted dmabuf heaps

- In fastrpc example [1] – dmabuf is allocated by userland from un-restricted heap.
- fastrpc kernel driver protects the buffer with QCOM SCM interface.
- Only kernel interface available to protect the buffer, hence can’t be done by userspace which is allocating the dmabuf.

Without restricted dmabuf heap, clients driver can protect the buffer using custom interface between userland but here client driver has secure usecase awareness.

- Restricted dmabuf heaps would keep the decision with userland about memory type for pipeline.

- [1] https://github.com/torvalds/linux/blob/master/drivers/misc/fastrpc.c#L817

Downstream Qualcomm Secure DMABUF Heaps

- Secure system heap
  - Allocation – system heap
  - Buffer protection – QCOM SCM assign
- Secure custom CMA heap
  - Allocation – custom cma heap[1]
  - Buffer protection – QCOM SCM assign
- Secure carveout heap
  - Allocation – carveout bitmap
  - Buffer protection – QCOM SCM assign

[1] https://lore.kernel.org/all/20231117100337.5215-1-quic_jasksing@quicinc.com/

Vendor DMABUF Heaps

- Taking example of restricted system heap
  - Non-protected DMABUF Heap - System heap
  - memory_protect/unprotect Methods
    - QCOM SCM assign
    - OPTEE/Vendor specific buffer protection

- Each vendor would need a separate heap based on memory protection method.
  - /dev/dma_heap/qcom,secure-pixel OR
  - /dev/dma_heap/vendor,optee

- [1] https://lore.kernel.org/linux-arm-kernel/20231111111559.8218-3-yong.wu@mediatek.com/
- [2] https://lore.kernel.org/linux-arm-kernel/91f0a8cf-3aef-4c54-b4b6-afd76cd5fdc8@quicinc.com/

Generic restricted heaps – Proposal under discussion

- As discussed earlier, vendors can have different buffer protection methods, but they can use common allocation methods from SG, CMA.
- [1] had discussed internal ops to allocate and secure restricted heap.
- Subsequently restricted_heap_ops and vendor restricted heaps using these ops are posted [2] [3] [4].

```
struct restricted_heap_ops {
        int     (*heap_init)(struct restricted_heap *rheap);
        int     (*alloc)(struct restricted_heap *rheap, struct restricted_buffer *buf);
        void    (*free)(struct restricted_heap *rheap, struct restricted_buffer *buf);
        int     (*restrict_buf)(struct restricted_heap *rheap, struct restricted_buffer *buf);
        void    (*unrestrict_buf)(struct restricted_heap *rheap, struct restricted_buffer *buf);
};
```


[1] https://lore.kernel.org/linux-arm-kernel/91f0a8cf-3aef-4c54-b4b6-afd76cd5fdc8@quicinc.com/

[2] https://lore.kernel.org/linux-arm-kernel/20240515112308.10171-6-yong.wu@mediatek.com/

[3] https://lore.kernel.org/linux-arm-kernel/20240720071606.27930-1-yunfei.dong@mediatek.com/#r

[4] https://lore.kernel.org/linux-arm-kernel/20240830070351.2855919-1-jens.wiklander@linaro.org/

Multiplexing heap names

restricted_heap_ops should allow code reuse for common allocation/buffer restriction methods, but this does add new named heaps per unique allocation/buffer protection combination.

Another possibility is if restricted named heaps represent usecase rather then exposing allocation/buffer restriction methods and fragmenting per vendor named heaps.

/dev/dma_heap/svp

QCOM secure video playback – building block

1. secure system heap [1] – Posted
2. Secure context bank support [2] – Posted, Note: This introduces method to support restricted buffer translation managed by HLOS arm-smmu driver.
3. Iris vidc non-secure support. [3] – Posted, Under Review
4. Iris vidc secure support – TBD

[1] https://lore.kernel.org/all/cover.1700544802.git.quic_vjitta@quicinc.com/

[2] https://lore.kernel.org/all/38274eb8-296e-c9c4-7eb1-232b852107cc@quicinc.com/

[3] https://patchwork.kernel.org/project/linux-media/list/?series=883741

Thank you

Follow us on:

For more information, visit us at qualcomm.com & qualcomm.com/blog