Revokable Grants

David Vrabel <david.vrabel@citrix.com>

Draft B

Contents

1 Introduction

1.1 Revision History

Version	Date	Changes
Draft A	24 Dec 2015	Initial draft.
Draft B	25 Jan 2015	Resolved some FIXMES: Don't allow GNTMAP_map_ref on revokable grants; allow up-to two mappings for each grant reference.

Version	Date	Changes
		New sub-sections on domain death and gntalloc driver notification mechanisms.
		Clarify that the revoke hypercall is only needed if the grant is in-use.

1.2 Purpose

Using grant references to share memory between mutually untrusting VMs is not possible because the grantee can keep grants mapped indefinitely. This limits the use of inter-domain communication mechanisms (such as libvchan) to between mutually trusting VMs.

This design proposes a mechanism where the VM granting access may specify that the grant is *revokable*. Access via such a grant may be revoked at any time, even if the grant is mapped by the other domain.

2 High Level Design

A revokable grant is indicated by an additional flag in the grant table entry. A domain may only map such a grant using a new sub-op (`GNTABOP_map_revokable`) and must supply a local GFN.

When the granting domain wishes to revoke a grant it:

1. Removes access from the grant, but does not make the grant available for other uses. The prevents any new grant map or copies from starting.
2. Makes a `GNTTABOP_revoke` hypercall if the grant is in use (e.g., mapped). The hypervisor atomically switches any mappings of the grant to the local GFN supplied when it was mapped. The hypervisor will also wait for any in-progress grant copies to complete.
3. Frees the grant references, making it available for other uses.

Grant mappers will need to handle grants being revoked, e.g., by copying the data from the shared page before checking the copied data is valid.

3 Low Level Design

3.1 Grant Table Entry

A new `GTF_revokable` flag is added. A grant reference with this bit set may only be mapped with `GNTTABOP_map_revokable` or copied with `GNTTABOP_grant_copy` (subject to the usual permission checks).

Attempts to use `GNTTABOP_map_grant_ref` with such a reference must fail with -EACCESS. Without a replacement page, revoking such a mapping would require clearing the mapping which would allow the granter to trigger faults in the mapper.

3.2 Hypercall ABI

Two new grant table sub-ops are added:

- `GNTTABOP_map_revokable`
- `GNTTABOP_revoke`

3.2.1 `GNTTABOP_map_revokable`

```
struct gnttab_map_revokable {
    struct gnttab_map_grant_ref map;
    xen_pfn_t lgfn;
};
```

Field	Purpose
`map`	Parameters as per the `GNTTAB_map_grant_ref` sub-op.
`lgfn`	A local GFN owned by the mapping domain that will be used if the grant is revoked.

The hypervisor will validate `lgfn` as owned by the mapping domain and take an additional reference. It will then map the grant reference as normal. The `lgfn` page will be recorded in the new map track entry.

A revokable grant may only be mapped twice. This limit is to prevent thousands of mappings causing performance problems for `GNTTABOP_revoke`. Two mappings are required for a PV guest to map a grant reference into both the kernel and userspace virtual address space.

3.2.2 GNTTABOP_revoke

```
struct gnttab_revoke {
    grant_ref_t ref;
};
```

Field	Purpose
ref	The grant reference whose access is being revoked.

The caller must first remove access from the grant reference to prevent any new grant maps or copies from starting.

For each mapping of this grant the hypervisor will atomically update the mapping to the local GFN in the map track entry. Both host and device (IOMMU) mappings will be updated. This ensures that the mapper will always see a valid mapping and will not receive unexpected page faults.

> FIXME: will probably need a way to efficiently walk the set of map track entries for a given grant ref. Scanning the whole map track table may be too slow.

Also, by taking the appropriate grant table lock, any grant copies will be known to be complete.

3.3 Domain Death

When a domain dies, all in-use revokable grants shall be revoked. This shall be done after other domains are no longer able to use grants from the dying domain (this is so the hypervisor does not need to clear the GTF_permit_access flag to block a grant's use).

3.4 Notification

libxenctrl provides mechanisms to notify mappers that the granting process has exited. These are implemented by IOCTL_GNTALLOC_SET_UNMAP_NOTIFY in Linux's gntalloc driver. The two mechanisms are:

- UNMAP_NOTIFY_CLEAR_BYTE – clear a byte in the shared page.
- UNMAP_NOTIFY_SEND_EVENT – send an event channel event.

`UNMAP_NOTIFY_CLEAR_BYTE` is not compatible with revokable grants. When the granting process dies the grant will be revoked and the mapper will see its local page, thus the cleared byte will not be visible.

`UNMAP_NOTIFY_SEND_EVENT` would be compatible but is unreliable because:

- Events are delivered asynchronously so the revoked mapping may be visible before the event is received.
- It only worked if the granting process dies. An event was not raised if the domain died.

The gntalloc driver shall return an EINVAL error if either option is requested for a revokable grant.