Disk Trimming

taradiddles ⋮ 9-11 minutes ⋮ 8/15/2023

Disk trimming is the procedure by which the operating system informs the underlying storage device of which storage blocks are no longer in use. It does this by issuing an `ATA_TRIM` command for the block. This is also known as a `discard`. In this way, the storage device can perform garbage collection of the unused blocks and internally prepare them for reuse. SSDs in general benefit from this, while HDDs do not.

In a Linux system running on bare metal, this is relatively straight-forward. When instructed by the operating system, discards are issued by the file-system driver directly to the storage driver and then to the SSD.

In Qubes, this gets more complex due to virtualization, LUKS, and LVM (and thin pools on R4.0 and up). If you run `fstrim --all` inside a TemplateVM, in a worst case the `discard` can follow a path like:

```
OS -> File-system Driver -> Virtual Storage Driver -> Backend Storage Driver -
> LVM Storage Driver -> LUKS Driver -> Physical Storage Driver -> Physical
Storage Device
```

If discards are not supported at any one of those layers, it will not make it to the underlying physical device.

There are some security implications to permitting TRIM (read for example this article), but in most cases not exploitable. Conversely, TRIM can improve security against local forensics when using SSDs, because with TRIM enabled deleting data (usually) results in the actual data being erased quickly, rather than remaining in unallocated space indefinitely. However deletion is not guaranteed, and can fail to happen without warning for a variety of reasons.

Configuration

In all versions of Qubes, you may want to set up a periodic job in `dom0` to trim the disk. This can be done with either systemd (weekly only) or cron (daily or weekly). You should also add this service or cron job to **each VM** you want the trim function to work in.

- **Systemd**

  From a terminal as a regular user:

  ```
  systemctl enable fstrim.timer
  systemctl start fstrim.timer
  ```

- **Cron**

  This can be done from a terminal as root, by creating a `trim` file in `/etc/cron.daily` (or `/etc/cron.weekly`). Add the following contents:

  ```
  #!/bin/bash
  /sbin/fstrim --all
  ```

  And mark it as executable with `chmod 755 /etc/cron.daily/trim`.

Note Although discards can be issued on every delete inside `dom0` by adding the `discard` mount option to `/etc/fstab`, this option can hurt performance so the above procedure is recommended instead. However,

inside App and Template qubes, the `discard` mount option is on by default to notify the LVM thin pool driver that the space is no longer needed and can be zeroed and re-used.

If you are using Qubes with LVM, you may also want to set `issue_discards = 1` in `/etc/lvm/lvm.conf`. Setting this option will permit LVM to issue discards to the SSD when logical volumes are shrunk or deleted. In R4.x, LVM Logical volumes are frequently deleted (every time a disposable VM is shut down, for example) so you may want to set `issue_discards = 1` if using an SSD, but see the article linked in the first section of this page. However, this is relatively rare in R3.x.

LUKS (R4.0)

If you have enabled LUKS in dom0, discards will not get passed down to the storage device in R4.0. Note that this procedure is no longer needed in R4.1, as discards are enabled by default.

To enable TRIM support in dom0 with LUKS you need to:

1. Get your LUKS device UUID:

   ```
   ls /dev/mapper/luks-*
   ```

2. Add entry to `/etc/crypttab` (replace luks-<UUID> with the device name and the <UUID> with UUID alone):

   ```
   luks-<UUID> UUID=<UUID> none discard
   ```

3. Add `rd.luks.options=discard` to kernel cmdline (follow either GRUB2 or EFI, not both):

   - GRUB2: `/etc/default/grub`, `GRUB_CMDLINE_LINUX` line and
     Rebuild grub config (`grub2-mkconfig -o /boot/grub2/grub.cfg`), then
     Rebuild initrd (`dracut -f`)
   - EFI: `/boot/efi/EFI/qubes/xen.cfg`, `kernel=` line(s), then
     Rebuild initrd (`dracut -f /boot/efi/EFI/qubes/initramfs-$(uname -r).img $(uname -r)`)

4. Reboot the system.

5. To verify if discards are enabled you may use `dmsetup table` (confirm the line for your device mentions “discards”) or just run `fstrim -av` (you should see a `/` followed by the number of bytes trimmed).

Swap Space

By default TRIM is not enabled for swap in both R4.0 and R4.1. To enable it add the `discard` flag to the options for the swap entry in `/etc/fstab`. This may or may not actually improve performance. If you only want the security against local forensics benefit of TRIM, you can use the `discard=once` option instead to only perform the TRIM operation once during at boot.

To verify that TRIM is enabled, check `dmesg` for what flags were enabled when the swap space was activated. You should see something like the following:

```
Adding 32391164k swap on /dev/mapper/qubes_dom0-swap.  Priority:-2 extents:1
across:32391164k SSDscFS
```

The `s` indicates that the entire swap device will be trimmed at boot, and `c` indicates that individual pages are trimmed after they are no longer being used.

▼ This document was migrated from the qubes-community project

- Page archive
- First commit: 08 Dec 2020. Last commit: 04 Jan 2022.
- Applicable Qubes OS releases based on commit dates and supported releases: 4.0
- Original author(s) (GitHub usernames): lubellier, awokd
- Original author(s) (forum usernames): @ludovic
- Document license: CC BY 4.0

tanky0u August 15, 2023, 1:20pm 2

Is this going to be the case in the QubesOS 4.2, too? Going forward, can we expect the disk trimming be enabled by default in QubesOS for SSD storages?

Johnboy December 27, 2023, 4:52am 3

R4.2 here.
rd.luks.options=discard is missing, but dmsetup table shows allow_discards for luks-uuid.
I guess it's working properly then?

Vlad May 10, 2024, 2:57pm 4

To see whether discard capability is enabled across your block device tree, you can run:

```
lsblk -D
```

solene May 10, 2024, 4:23pm 5

There is a LVM knob to turn on in /etc/lvm.conf IIRC otherwise the trim can't be done.

Vlad August 18, 2024, 9:27pm 6

> There is a LVM knob to turn on in /etc/lvm.conf

Correction: `issue_discards` in `/etc/lvm/lvm.conf`

You need to set `issue_discards = 1` there, as per original post.

Vlad August 19, 2024, 2:19am 7

Not for me.

@taradiddles - doesn't hold for me in Qubes 4.2.2. I had to enable all 3 items to get discards working in dom0:

- `issue_discards = 1` in `/etc/lvm/lvm.conf`
- `none discard` in `/etc/crypttab`
- add `rd.luks.options=discard` to kernel cmdline

fsflover December 8, 2024, 4:08pm 8

@adw isn't this a bug/regression in R4.2?

solene December 9, 2024, 9:52am 9

check `lsblk --discard` output in dom0.

> And check the values of DISC-GRAN (discard granularity) and DISC-MAX (discard max bytes) columns. Non-zero values indicate TRIM support.

source Arch Linux wiki: Solid state drive #trim