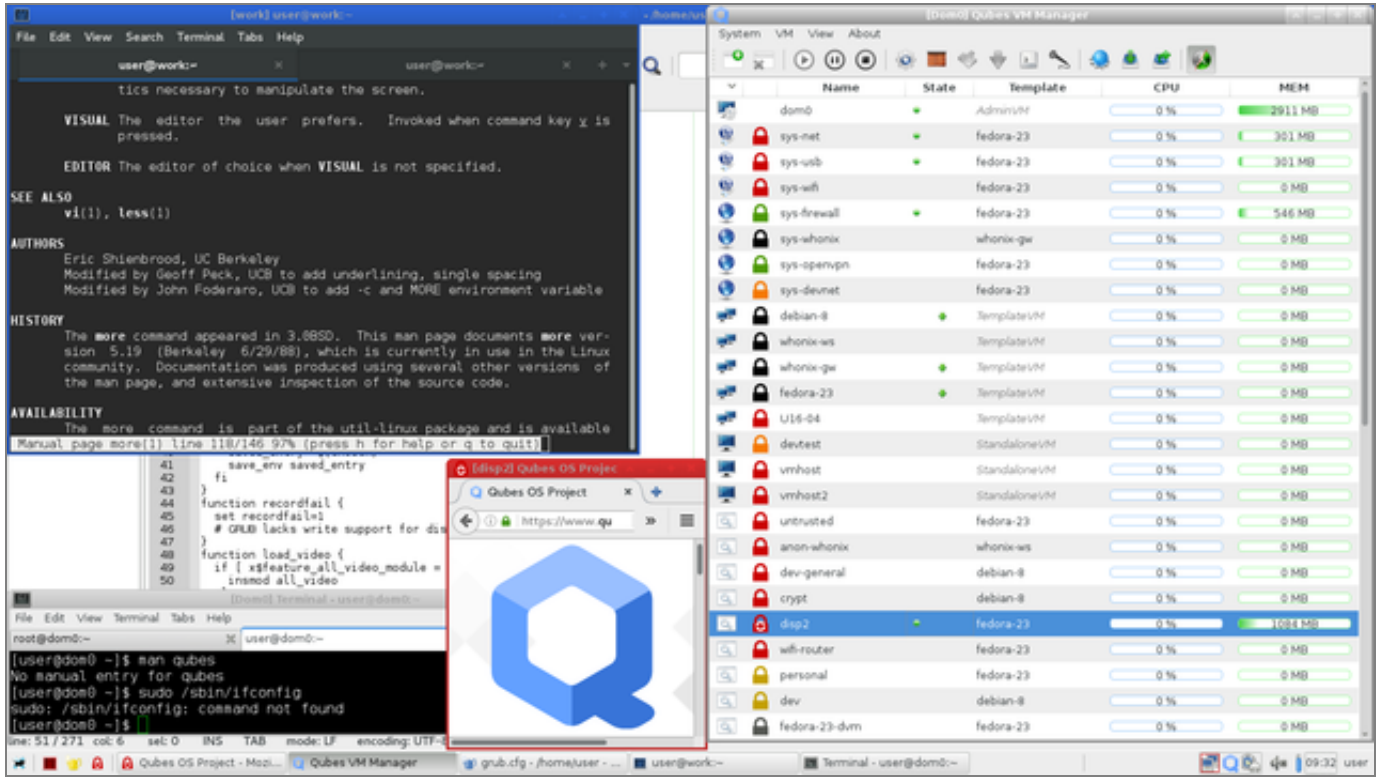
Qubes OS

Virtualize Everything

Qubes (https://www.qubes-os.org/) is a "security-oriented operating system" built atop the Xen (https://xenproject.org/) hypervisor (https://en.wikipedia.org/wiki/Hypervisor). Major functionality, including networking and user applications, is split across separate Virtual Machines (https://en.wikipedia.org/wiki/Virtual_machine) ("domains" in xen-speak) which Qubes aims to securely integrate/orchestrate. The technical details are very interesting, and we'll get to some of them, but the first question is whether Qubes is ready for real-world use.



(/static/images/20160812-Qubes/standard_use.png)

The screenshot shows a fairly typical, if busy, system (Qubes also provides much nicer screenshots (https://www.qubes-os.org/screenshots/)). The Qubes VM Manager (right side) shows all the system domains including their type, security domain, name, status (green=running), template, and resource usage. The bottom left terminal is executing in the `dom0` domain (which has no network interfaces for `/sbin/ifconfig` anyhow). The top left terminal is executing "man (https://en.wikipedia.org/wiki/Man_page) vi (https://en.wikipedia.org/wiki/Vi)" in the `work` domain. A disposable VM (https://www.qubes-os.org/doc/dispvm/) is executing Firefox (https://en.wikipedia.org/wiki/Firefox_%28browser%29) in the bottom center. Geany (https://en.wikipedia.org/wiki/Geany) is showing some lines from an unrelated grub.cfg (https://en.wikipedia.org/wiki/GNU_GRUB) in the background (executed in the `work` domain).

Overall Thoughts

Applications in Qubes ran pretty normal on the whole and overall speed felt unchanged. While VM startup some time (15-30s), VMs were generally responsive/fast thereafter (surprising given all the behind-the-scenes VM interactions). One "fix" to these startup times is to "autostart" commonly used domains on boot. This mostly solves the issue, except for disposable VMs (https://www.qubes-os.org/doc/dispvm/) which are created/destroyed on demand.

The learning curve to use/customize Qubes was fairly steep, but implementation choices felt reasonable and well thought out. It still took a lot of reading and some getting used to however. For example, the VM templating system (https://www.qubes-os.org/doc/template-implementation/) meant most VMs could only permanently modify specific locations, which was sometimes surprising. Security domains also didn't feel very natural, but I haven't found the right balance between one-VM-for-everything and one-VM-per-application. Most critically in my case, my thousands of VirtualBox (https://en.wikipedia.org/wiki/VirtualBox) VMs used for local development and testing don't play well with Xen (https://en.wikipedia.org/wiki/Xen).

Overall, Qubes OS made some very smart architectural choices that appear to add strong protections. It didn't seem very well suited to my specific (development focused) workflow (covered more below), but I was very pleased with the ease of use and will continue using it on my laptop. In an ideal world (wherein I eventually escape the VirtualBox lock-in, resolve security domain overuse, and fix several less important quality of life issues) Qubes OS may make a great next OS. For security minded individuals with more standard environments, Qubes OS may be the right OS even now.

System Setup

Download Qubes Release 3.1 (https://www.qubes-os.org/downloads/) and verify the signature (https://www.qubes-os.org/doc/verifying-signatures/) as below. Frustratingly, it doesn't appear possible to verify the Release Signing Key without ultimately trusting the Master Signing Key. `--list-sig` appears to verify at least the short key id, but that is not secure (https://evil32.com/).

```
wget https://mirrors.kernel.org/qubes/iso/Qubes-R3.1-x86_64.iso
wget https://mirrors.kernel.org/qubes/iso/Qubes-R3.1-x86_64.iso.asc
gpg --keyserver pool.sks-keyservers.net --recv-keys 0x427F11FD0FAA4B080123F01CDDFA1A3E36879494 # Qubes Master Signing Key
gpg --keyserver pool.sks-keyservers.net --recv-keys 0xC52261BE0A823221D94CA1D1CB11CA1D03FA5082 # Release 3 Signing Key

gpg --verbose --verify Qubes-R3.1-x86_64.iso.asc Qubes-R3.1-x86_64.iso
# gpg: armor header: Version: GnuPG v1
# gpg: Signature made Tue 08 Mar 2016 10:40:56 PM EST using RSA key ID 03FA5082
# gpg: using PGP trust model
# gpg: Good signature from "Qubes OS Release 3 Signing Key"
# gpg: WARNING: This key is not certified with a trusted signature!
# gpg:          There is no indication that the signature belongs to the owner.
# Primary key fingerprint: C522 61BE 0A82 3221 D94C  A1D1 CB11 CA1D 03FA 5082
# gpg: binary signature, digest algorithm SHA256

gpg --list-sig 0xC52261BE0A823221D94CA1D1CB11CA1D03FA5082
# ----------------------------
# pub   4096R/36879494 2010-04-01
# uid                  Qubes Master Signing Key
# ...
# sig 3        36879494 2010-04-01  Qubes Master Signing Key
#
# pub   4096R/03FA5082 2014-11-19
# uid                  Qubes OS Release 3 Signing Key
# sig          36879494 2014-11-19  Qubes Master Signing Key
# sig 3        E2986940 2016-01-04  [User ID not found]
# sig 3        03FA5082 2014-11-19  Qubes OS Release 3 Signing Key
```

Installation was straightforward even with preexisting software RAID (https://en.wikipedia.org/wiki/Mdadm) and disk encryption (https://en.wikipedia.org/wiki/Linux_Unified_Key_Setup). Everything basically worked out of the box. I choose to install Xfce (https://en.wikipedia.org/wiki/Xfce) instead of KDE (https://en.wikipedia.org/wiki/KDE), which no longer requires [a] notable amount of manual configuration (https://www.qubes-os.org/doc/xfce/), and the system started without any headaches. Very impressive.

Technical Commentary

Qubes OS gave a very promising experience on the whole, but many non-critical technical issues popped up all the same. These are covered in no particular order below:

1. Passwordless root access (ITL rationale (https://www.qubes-os.org/doc/vm-sudo/)): Despite the excellent write-up, this isn't a clear win, but security products often need to simplify user life wherever possible for traction. ITL already provides a blueprint for an enhanced alternative dom0-prompt (https://www.qubes-os.org/doc/vm-sudo/#replacing-password-less-root-access-with-dom0-user-prompt) though; kudos for that.

2. AppVMs (https://www.qubes-os.org/getting-started/) use TemplateVM (https://www.qubes-os.org/doc/template-implementation/) data with per-AppVM customization in /home, /usr/local, and /rw/config directories (everything else is "erased" on AppVM shutdown). This means a user can `sudo apt-get install vim` and use vim for that specific AppVM instance, but it will be gone after an AppVM restart. This is really just an initial headache though; the workaround is either installing the application in the TemplateVM or forcing installation into `/usr/local/bin` (which is prepended to the PATH (https://kb.iu.edu/d/acar)).

3. Configuring/security Firefox in the disposable VMs (https://www.qubes-os.org/doc/dispvm/): The documentation (https://www.qubes-os.org/doc/dispvm-customization/) seems straightforward in hindsight, but took several attempts to walk through correctly. In my first attempt to install browser add-ons (NoScript (https://en.wikipedia.org/wiki/NoScript), HTTPS Everywhere (https://www.eff.org/https-everywhere), etc), I manually connected the fedora-23 TemplateVM to the firewall (a **big** no no) and endangered every NetVM/ProxyVM/AppVM based on the fedora-23 TemplateVM.

4. Qubes supports copy and paste between domains (https://www.qubes-os.org/doc/copy-paste/) with Ctrl+Shift+C and Ctrl+Shift+V, but regular users of the terminal will recognize that as copy and paste in the terminal (http://howtoubuntu.org/how-to-cut-copy-and-paste-in-the-terminal-in-ubuntu). Fortunately, the `secure_copy_sequence` can be changed in `/etc/qubes/guid.conf`, but the Super/Windows/Meta/Mod4 key never seemed to work on my systems. On the other hand, copying files between domains (https://www.qubes-os.org/doc/copying-files/) worked great!

5. `sys-usb` did not play nice with my (USB) keyboard/mouse KVM. I could partially workaround this issue by only assigning the "other" USB hub to `sys-usb`, but that left the standard USB threat via KVM intact. I suspect the USB Mass Storage devices plugged into `sys-usb` could be shared out to other VMs in some manner, but manually copying/moving (https://www.qubes-os.org/doc/copying-files/) files was a serious headache. Fortunately, this appears to be partially mitigated in Qubes 3.2 (https://www.qubes-os.org/doc/usb/#attaching-a-single-usb-device-to-a-qube-usb-passthrough).

6. Selecting/deselecting applications in VM settings resulted in a fairly slow (10s+) update process that made initial setup fairly frustrating. The (bottom left) Application Menu grouped applications into a submenu for each VM (e.g. `Domain: personal` VM with submenu items `[Add more shortcuts, Files, Firefox, Geany, Terminal]`), but right clicking the VM didn't provide a similar

(VM-specific) application menu.

7. The above was only an issue because 9 times out of 10, when I clicked on a VM, I wanted to start a terminal. The fastest way to do it across all my (excessive) domains was right click -> "Run command in VM" -> type "gnome-terminal" (got old very quickly). Going one step further, a `Restart VM` option under `Pause VM` would be great to simplify restarts after TemplateVM updates.
8. Qubes VM Manager doesn't currently provide VM grouping. This isn't technically a flaw in Qubes OS, but made managing large numbers of VMs very difficult.
9. Instancing disposable VMs (https://www.qubes-os.org/doc/dispvm/) took 15-30s (and much longer after updating the template `fedora-23-dvm`). This delay proved very disruptive during normal tasks. Long-term this could be resolved by keeping a small cache of "warm" disposable VMs (ready and idle) to offload initial startup into an asynchronous/backend system.
10. VMs (including NetVMs and ProxyVMs) may only "connect" to one other VM. In the image above, this means `sys-firewall` can either connect to `sys-net` for ethernet access or `sys-wifi` for wireless access, but can't select between them for routing.

VM Types

Type	Description
AdminVM	dom0 (https://wiki.xen.org/wiki/Dom0) (the only AdminVM) manges VM interactions and (until 4.1 (https://github.com/rootkovska/qubes-roadmap)) runs the GUI subsystem
NetVM	Device-specific VMs that directly interact with (untrusted) hardware for the system (Ethernet, Wifi, USB)
ProxyVM	Manage network connections between virtual machines (sys-firewall, sys-openvpn, sys-whonix)
TemplateVM	Enables thin virtual machines in other domains; provides system packages / configuration (Fedora 23 and Debian 8 by default)
StandaloneVM	Full virtual machines with persistent package management (requires manual updating)
AppVM	Standard user applications run in these. AppVMs execute TemplateVMs (with customization available)
DispVM	Disposable Virtual Machines generated, then destroyed, for a single purpose (templated from `fedora-23-dvm`)
HardwareVM	Qubes added Windows support (http://blog.invisiblethings.org/2012/12/14/qubes-2-beta-1-with-initial-windows.html) through custom Xen HVM (https://en.wikipedia.org/wiki/Xen#Hardware-assisted_virtualization.2C_allowing_for_unmodified_guests), but HardwareVMs allow running "any" OS inside Qubes OS

System VMs

Name	Type	Descrption
dom0	AdminVM	Coordinates and orchestrates Qubes OS
sys-usb	NetVM	Directly interfaces with USB devices. May be configured to allow USB keyboard/mouse passthru.
sys-net	NetVM	Directly interfaces with network devices by default. I reconfigured to only manage ethernet devices.
sys-wifi	NetVM	Directly interfaces with wireless devices. Only accessed by `sys-openvpn` due to low-trust wireless networks.
sys-firewall	ProxyVM	Manually connected to sys-net (ethernet) or sys-openvpn (wifi) based on network availability/needs.
sys-openvpn	ProxyVM	Connects to sys-wifi to provide an OpenVPN (https://en.wikipedia.org/wiki/OpenVPN) tunnel for any VMs connecting to sys-openvpn.
sys-whonix	ProxyVM	Connects to sys-firewall to provide an anonymous Tor (https://en.wikipedia.org/wiki/Tor_%28anonymity_network%29) tunnel for any VMs connecting to sys-whonix.
sys-devnet	ProxyVM	Connects to sys-firewall providing development networking from `dev-general` VM to active development VMs.
debian-8	TemplateVM	Qubes customized Debian 8 (https://en.wikipedia.org/wiki/Debian) installation. Don't connect to internet (update via "Update VM")
fedora-23	TemplateVM	Qubes customized Fedora 23 (https://en.wikipedia.org/wiki/Fedora) installation. Template for most system VMs. Don't connect to internet.
whonix-gw	TemplateVM	Qubes customized Whonix Gateway (https://www.qubes-os.org/doc/whonix/). Template for `sys-whonix` ProxyVM.
whonix-ws	TemplateVM	Qubes customized Whonix Workstation (https://en.wikipedia.org/wiki/Whonix). Template for `anon-whonix` AppVM.