Secure Desktops with Qubes: Compartments

KYLE RANKIN

Kyle Rankin is a Sr. Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

Figuring out how to compartmentalize your desktop across VMs can be daunting, so I've provided an example of how I do it to help you get started.

PREVIOUS
Dave Taylor's Work the Shell

NEXT
Shawn Powers' The Open-Source Classroom

THIS IS THE THIRD ARTICLES IN MY SERIES ABOUT QUBES. In the first two articles, I gave an overview about what Qubes is and described how to install it. One of the defining security features of Qubes is how it lets you compartmentalize your different desktop activities into separate VMs. The idea behind security by compartmentalization is that if one of your VMs is compromised, the damage is limited to just that VM.

When you first start using Qubes, you may not be quite sure how best to divide up all of your files and activities into separate VMs. I know when I first started using it, I found inspiration in Joanna Rutkowska's (Qubes' creator) paper on how she used Qubes (http://invisiblethingslab.com/resources/2014/Software_compartmentalization_vs_physical_separation.pdf). In this article, I describe how I organize my activities into VMs on my personal computer. Although I'm not saying my approach is perfect, and I certainly could secure things even further than I do, I at least will provide you one example you can use to get started.

Summary of Qubes Concepts

In my previous article, I elaborated on overall Qubes concepts like the different VM types, trust levels and other features, but since I refer to those concepts in this article as well, here's a brief summary. (If you want to know more, read my column in the April and May 2016 issues.)

The first concept to understand with Qubes is that it groups VMs into different categories based on their use. Here are the main categories of VMs I refer to in the rest of the article:

- Disposable VM: these also are referred to as dispVMs and are designed for one-time use. All data in them is erased when the application is closed.

- Domain VM: these also often are referred to as appVMs. They are the VMs where most applications are run and where users spend most of their time.

- Service VM: service VMs are split into subcategories of netVMs and proxyVMs. These VMs typically run in the background and provide your appVMs with services (usually network access).

- Template VM: other VMs get their root filesystem template from a Template VM, and once you shut the appVM off, any changes you may have made to that root filesystem are erased (only changes in /rw, /usr/local and /home persist). Generally, Template VMs are left powered off unless you are installing or updating software.

When you create new VMs of any type, you can assign them a color based on your level of trust on a continuum from red (untrusted) to orange and yellow to green (somewhat more trusted) to blue and purple and grey (even more trusted) to black (ultimately trusted). The window borders and icons for a particular VM are colorized based on their trust level, so you get visual cues that help prevent you from, for instance, pasting trusted passwords into an untrusted VM.

Although by default all new Qubes VMs you create have unlimited network access, Qubes allows you to create firewall rules to restrict what a VM can do. If your VM doesn't need network access (such as for the highly-trusted vault VM you can use to store GPG keys and password vaults), you even can remove the network device completely.

In a default install, Qubes provides a few appVMs to help you get started:

- untrusted appVM: red.
- personal appVM: yellow.
- work appVM: green.
- vault appVM: black.

The idea is for you to perform any general-purpose untrusted activities (like general Web browsing) in the untrusted VM and not store any personal files there. Then you can perform more trusted activities like checking your e-mail or any Web browsing that requires personal credentials in the personal VM. You can check your work e-mail and store your work documents in the work VM. Finally, you can store your GPG keys and password manager files in the vault (which has no network at all). Although this is nice for getting started, as you can see, you may want to isolate your activities and files even further.

The installer also creates a sys-net, sys-firewall and sys-whonix service VM to provide you with network access, a firewall for appVMs and a Tor gateway, respectively. You also optionally can enable a sys-usb service VM that is assigned all of your USB controllers to protect the rest of the

system from USB-based attacks.

My Personal Computer

My personal computer is a Purism Librem 13, and my general desktop use is pretty basic. Here's my normal list of activities in order of risk:

- Web browsing.
- Checking e-mail.
- Chatting on IRC.
- Using my 3D printer.
- Writing articles.

Generally speaking, Web browsing and e-mail are the riskiest activities I perform on my computer each day, as they can expose me to malicious file attachments and other compromises. On the other end, all I need to write articles is a text editor with no network access, so that's a pretty safe activity. Below I list the different appVMs I've created based on this type of use, ordered from least-trusted to most-trusted. I also show what color I assigned the VM and describe how I use each appVM.

dispVM—red: I use disposable VMs whenever I'm doing something particularly risky, such as when I want to view a sketchy-looking URL. For instance, my mail client is configured to open all attachments automatically in a disposable VM (based on the official Qubes mutt guide: https://www.qubes-os.org/doc/mutt). That way, even if someone were to send me a malicious Word document or PDF, I can read it in the disposable VM, and the attack is isolated inside that VM. When I close the document, any malicious program it is running goes away and in the meantime, the attacker had no access to any of my personal files.

untrusted—red: My untrusted appVM is where I perform all of my general-purpose Web browsing but not any Web sites that require a user name and password. It has unrestricted access to the Internet. I've set up some other more-trusted VMs (such as the one where I chat in IRC)

In this way, any tracking cookies are limited to the Web browser inside this appVM; Tor prevents any other servers apart from Facebook from knowing I'm using Facebook, and even Facebook itself doesn't know my IP.

to open up URLs in this VM automatically (by setting the default Web browser in that appVM to be the qvm-open-in-vm command-line tool). I don't store any personal files in my untrusted VM, so if I feel like a URL I opened looks particularly sketchy, I can just delete the VM and re-create it, and in less than a minute, I'm back with a clean untrusted VM.

Since I browse random Web sites with this VM and might open obscured URL-shortened URLs in it, it's one of the VMs most likely to be compromised. That said, because I don't store any personal files in the VM, and I don't browse to any Web sites that require a user name and password, the most an attacker could do besides just use that VM for its network and CPU resources is view my general browsing habits.

fb—orange: It may surprise some readers to know that I have a Facebook account. I personally don't post to my account all that much (and when I do, I post only things I'm fine with the whole world seeing), but like many of you, I have friends that I don't see often who post about what's going on with their lives only on Facebook. I'm concerned about the privacy issues surrounding Facebook tracking my every move on the Web, but I still want to be able to view my friends' posts, which I can't do without logging in.

My compromise has been to create a special appVM just for Facebook and nothing else. This appVM is configured to use the sys-whonix proxyVM for network access, so all of its traffic goes over Tor, and I use Facebook's Tor hidden service at https://facebookcorewwwi.onion to access the site. In this way, any tracking cookies are limited to the Web browser inside this appVM; Tor prevents any other servers apart from Facebook from knowing I'm using Facebook, and even Facebook itself doesn't know my IP.

personal-web—yellow: Because Web browsing is one of the riskier activities one can perform, I've decided to separate my authenticated Web browsing not only from my general Web browsing, but also from the rest of my personal files. Since sites that provide a login also usually let you log in over HTTPS, I restrict this VM's network access so it can connect only to port 443 on the Internet. The personal-web appVM is set aside for any site that needs a user name and password (outside of banking). So, for instance, when I'm shopping on-line, I might use my untrusted VM, but when I'm ready to log in and buy something, I use my personal-web appVM. I've set up my password vault to open URLs in this appVM automatically instead of the untrusted one.

The idea here is to prevent an attacker who has compromised my untrusted appVM through a malicious Web site from being able to grab any of my Web credentials. Although it's true that attackers who compromise one of the many Web sites I log in to through personal-web would be able to get credentials for other sites, they still wouldn't be able to access any of my personal files (like documents or GPG or SSH keys). They also wouldn't be able to access my banking, because I separate my banking into its own VM. Some Qubes users who are concerned about this sort of thing end up launching disposable VMs for any authenticated sessions.

personal—yellow: My personal appVM is the closest to a traditional user's home directory, and it contains the bulk of my personal files, such as my SSH keys. That said, I don't do any Web browsing from this appVM and use either the untrusted, personal-web or finance VMs for that. I mostly use this VM to `ssh` to other servers I manage, check e-mail, connect to a remote screen session that I use for IRC and manage a few personal GitHub projects. Because of this, I can restrict the Qubes firewall so it allows only outbound SSH to any remote IP, and otherwise to open only the handful of ports IMAP and SMTP need explicitly to my mail server.

Because this VM contains the bulk of my personal files and my (password-protected) SSH key, I'm more careful about what I do in this VM than in some of the others. That's a big reason why I don't browse the Web from this VM and why although I check e-mail from this VM, I automatically open all attachments in a disposable VM.

Qubes provides a service known as split-GPG that acts like a GPG wrapper program you can use in other appVMs whenever you want to access a GPG key inside a vault.

printrbot—green: I decided to set aside a special appVM just for interacting with my printrbot, because I have a shared internal network volume I mount that stores all of my 3D models. This VM controls my 3D printer and also launches the Cura program I use to modify and slice 3D models. By splitting this off to its own appVM, I can use the Qubes firewall to restrict access to just my local Octoprint server and my network storage, since this VM has no need to use the Internet at large.

finance—green: Because of how sensitive a person's financial accounts are compared to other accounts, I've decided to create a special appVM that's normally off, that I use only for banking. That way, my banking credentials won't leak to other appVMs. This VM is locked down so that it can connect only to port 443 on the Internet at large, and if I wanted to lock it down further, I could restrict that to only the hostnames belonging to my banking sites.

writing—blue: This VM is the VM I'm using right now, because it is where I store all of my personal documents and organize my writing. I decided to isolate this activity from others partially because I don't want to risk leaking books I'm working on that are unreleased, but also because I realized this VM really needs no network apart from access to a local git repository I use to organize a few writing projects, so I can create really restrictive firewall rules.

vault—black: The vault is the most sensitive and most trusted appVM in my environment. Instead of just creating restrictive firewall rules for this host, to make it as secure as possible, it has no network device at all. I use this VM to store my GPG key and my KeepassX password vault. Qubes provides a service known as split-GPG that acts like a GPG wrapper program you can use in other appVMs whenever

you want to access a GPG key inside a vault. Basically, an appVM that wants access to the key sends its encryption or decryption payload to the vault VM using the Qubes wrapper script. You get a colorized prompt on your desktop asking whether you want to allow the appVM to have access to the vault's GPG key for a set period of time. If you accept, the payload goes to the vault, the vault encrypts or decrypts it, and then the output goes back to the appVM. In that way, your appVM never sees the GPG private key in the vault, and it behaves kind of like a poor-man's Hardware Security Module.

I hope that seeing how I organize my Qubes desktop will help you figure out how best to organize your own VMs. Generally speaking, it comes down to separating different types of activities and files from each other based on risk. The guiding principle is to assume it's possible to break into a particular VM (in particular, less-trusted VMs) and to try to limit the files and access any certain VM has. If you want some more specific guides on Qubes best practices, the first place to start is Qubes' own documentation page (https://www.qubes-os.org/doc) as it provides a number of useful guides for common activities (like setting up split-GPG).■

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

RETURN TO CONTENTS