October 15, 2018

Flatpaks, sandboxes and security

Last week the Flatpak community woke to the "news" that we are making the world a less secure place and we need to rethink what we're doing. Personally, I'm not sure this is a fair assessment of the situation. The "tl;dr" summary is: Flatpak confers many benefits besides the sandboxing, and even looking just at the sandboxing, improving app security is a huge problem space and so is a work in progress across multiple upstream projects. Much of what has been achieved so far already delivers incremental improvements in security, and we're making solid progress on the wider app distribution and portability problem space.

Sandboxing, like security in general, isn't a binary thing – you can't just say because you have a sandbox, you have 100% security. Like having two locks on your front door, two front doors, or locks on your windows too, sensible security is about defense in depth. Each barrier that you implement precludes some invalid or possibly malicious behaviour. You hope that in total, all of these barriers would prevent anything bad, but you can never really guarantee this – it's about multiplying together probabilities to get a smaller number. A computer which is switched off, in a locked faraday cage, with no connectivity, is perfectly secure – but it's also perfectly useless because you cannot actually use it. Sandboxing is very much the same – whilst you could easily take systemd-nspawn, Docker or any other container technology of choice and 100% lock down a desktop app, you wouldn't be able to interact with it at all.

Network services have incubated and driven most of the container usage on Linux up until now but they are fundamentally different to desktop applications. For services you can write a simple list of permissions like, "listen on this network port" and "save files over here" whereas desktop applications have a *much* larger number of touchpoints to the outside world which the user expects and requires for normal functionality. Just thinking off the top of my head you need to consider access to the filesystem, display server, input devices, notifications, IPC, accessibility, fonts, themes, configuration, audio playback and capture, video playback, screen sharing, GPU hardware, printing, app launching, removable media, and joysticks. Without making holes in the sandbox to allow access to these in to your app, it either wouldn't work at all, or it wouldn't work in the way that people have come to expect.

What Flatpak brings to this is understanding of the specific desktop app problem space – most of what I listed above is to a greater or lesser extent understood by Flatpak, or support is planned. The Flatpak sandbox is very configurable, allowing the application author to specify which of these resources they need access to. The Flatpak CLI asks the user about these during installation, and we provide the `flatpak override` command to allow the user to add or remove these sandbox escapes. Flatpak has introduced portals into the Linux desktop ecosystem, which we're really pleased to be sharing with snap since earlier this year, to provide runtime access to resources outside the sandbox based on policy and user consent. For instance, document access, app launching, input methods and recursive sandboxing ("sandbox me harder") have portals.

The starting security position on the desktop was quite terrible – anything in your session had basically complete access to everything belonging to your user, and many places to hide.

- Access to the X socket allows arbitrary input and output to any other app on your desktop, but without it, no app on an X desktop would work. Wayland fixes this, so Flatpak has a fallback setting to allow Wayland to be used if present, and the X socket to be shared if not.
- Unrestricted access to the PulseAudio socket allows you to reconfigure audio routing, capture microphone input, etc. To ensure user consent we need a portal to control this, where by default you can play audio back but device access needs consent and work is under way to create this portal.
- Access to the webcam device node means an app can capture video whenever it wants – solving this required a whole new project.
- Sandboxing access to configuration in dconf is a priority for the project right now, after the 1.0 release.

Even with these caveats, Flatpak brings a bunch of default sandboxing – IPC filtering, a new filesystem, process and UID namespace, seccomp filtering, an immutable /usr and /app – and each of these is already a barrier to certain attacks.

Looking at the specific concerns raised:

- Hopefully from the above it's clear that sandboxing desktop apps isn't just a switch we can flick overnight, but what we already have is far better than having nothing at all. It's not the intention of Flatpak to somehow mislead people that sandboxed means somehow impervious to all known security issues and can access nothing whatsoever, but we do want to encourage the use of the new technology so that we can work together on driving adoption and making improvements together. The idea is that over time, as the portals are filled out to cover the majority of the interfaces described, and supported in the major widget sets / frameworks, the criteria for earning a nice "sandboxed" badge or submitting your app to Flathub will become stricter. Many of the apps that access `--filesystem=home` are because they use old widget sets like Gtk2+ and frameworks like Electron that don't support portals (yet!). Contributions to improve portal integration into other frameworks and desktops are very welcome and as mentioned above will also improve integration and security in other systems that use portals, such as snap.

- As Alex has already blogged, the freedesktop.org 1.6 runtime was something we threw together because we needed something distro agnostic to actually be able to bootstrap the entire concept of Flatpak and runtimes. A confusing mishmash of Yocto with flatpak-builder, it's thankfully nearing some form of retirement after a recent round of security fixes. The replacement freedesktop-sdk project has just released its first stable 18.08 release, and rather than "one or two people in their spare time because something like this needs to exist", is backed by a team from Codethink and with support from the Flatpak, GNOME and KDE communities.
- I'm not sure how fixing and disclosing a security problem in a relatively immature pre-1.0 program (in June 2017, Flathub had less than 50 apps) is considered an ongoing problem from a security perspective. The wording in the release notes?

Zooming out a little bit, I think it's worth also highlighting some of the other reasons why Flatpak exists at all – these are far bigger problems with the Linux desktop ecosystem than app security alone, and Flatpak brings a huge array of benefits to the table:

- **Allowing apps to become agnostic of their underlying distribution**. The reason that runtimes exist at all is so that apps can specify the ABI and dependencies that they need, and you can run it on whatever distro you want. Flatpak has had this from day one, and it's been hugely reliable because the sandboxed /usr means the app can rely on getting whatever they need. This is the foundation on which everything else is built.
- **Separating the release/update cadence of distributions from the apps**. The flip side of this, which I think is huge for more conservative platforms like Debian or enterprise distributions which don't want to break their ABIs, hardware support or other guarantees, is that you can still get new apps into users hands. Wider than this, I think it allows us huge new freedoms to move in a direction of reinventing the distro – once you start to pull the gnarly complexity of apps and their dependencies into sandboxes, your constraints are hugely reduced and you can slim down or radically rethink the host system underneath. At Endless OS, Flatpak literally changed the structure of our engineering team, and for the first time allowed us to develop and deliver our OS, SDK and apps in independent teams each with their own cadence.
- **Disintermediating app developers from their users**. Flathub now offers over 400 apps, and (at a rough count by Nick Richards over the summer) over half of them are directly maintained by or maintained in conjunction with the upstream developers. This is fantastic – we get the releases when they come out, the developers can choose the dependencies and configuration they need – and they get to deliver this same experience to everyone.
- **Decentralised**. Anyone can set up a Flatpak repo! We started our own at Flathub because there needs to be a center of gravity and a complete story to build out a user and developer base, but the idea is that anyone can use the same tools that we do, and publish whatever/wherever they want. GNOME uses GitLab CI to publish nightly Flatpak builds, KDE is setting up the same in their infrastructure, and Fedora is working on completely different infrastructure to build and deliver their packaged applications as Flatpaks.
- **Easy to build**. I've worked on Debian packages, RPMs, Yocto, etc and I can honestly say that flatpak-builder has done a very good job of making it really easy to put your app manifest together. Because the builds are sandboxed and each runtimes brings with it a consistent SDK environment, they are very reliably reproducible. It's worth just calling this out because when you're trying to attract developers to your platform or contributors to your app, hurdles like complex or fragile tools and build processes to learn and debug all add resistance and drag, and discourage contributions. GNOME Builder can take any flatpak'd app and build it for you automatically, ready to hack within minutes.
- **Different ways to distribute apps**. Using OSTree under the hood, Flatpak supports single-file app .bundles, pulling from OSTree repos and OCI registries, and at Endless we've been working on peer-to-peer distribution like USB sticks and LAN sharing.

Nobody is trying to claim that Flatpak solves all of the problems at once, or that what we have is anywhere near perfect or completely secure, but I think what we have is pretty damn cool (I just wish we'd had it 10 years ago!). Even just in the security space, the overall effort we need is huge, but this is a journey that we are happy to be embarking together with the whole Linux desktop community. Thanks for reading, trying it out, and lending us a hand.

posted by ramcq @ 1:40 pm
Comments (5) .:. Trackback .:. Permalink

5 responses to "Flatpaks, sandboxes and security"

1. *loupianche* says:
   October 15, 2018 at 4:48 pm

   For flathub, is there any security team? What happens when a flatpak has a serious security issue?

   Reply

   - *ramcq* says:
     October 15, 2018 at 9:10 pm

     > For flathub, is there any security team? What happens when a flatpak has a serious security issue?

     Not yet/officially, no. In principle each app submitter/maintainer is responsible for the security issues within their own Flatpak. However, one of the core Flathub team, Patrick Griffis, has been working on an automated CVE checking tool. The plan is it can run across all of the apps and runtimes to highlight components with potentially known vulnerabilities. You can see recent output for the main runtimes for

example.

Reply

2. *Sunny Sigara* says:
October 15, 2018 at 8:12 pm

Just after the websites exposed it's flaw, a immediate reassurance article came out by the dev who's previous article was in July 29, 2017!!

Wow!

Reply

- *ramcq* says:
October 15, 2018 at 9:29 pm

> Just after the websites exposed it's flaw, a immediate reassurance article came out by the dev who's previous article was in July 29, 2017!!
>
> Wow!

Well, I never said I was a dev – I'm actually just a very geeky manager. I'm an active member of the Flatpak community, and I am the main sysadmin for Flathub. In my day job at Endless we were the first to deploy Flatpak in production in 2016, and I've spoken about Flatpak and Flathub this year at devconf.cz and GUADEC.

Reply

- *Jonathan* says:
October 16, 2018 at 8:33 am

If that website had been published as a reddit comment, or a random blogpost on blogger, or wordpress, (which is about the calibre of writing, authority and expertise we're talking about here), it wouldn't have got any attention, it would have just been another misinformed rant into the noise. It only got any press because somebody paid $5/year or whatever to point a domain at it.

Rob has nevertheless spent the time to engage the points raised in that drivel, and rather than engage with the points raised within you're going to question its relevance because his last blog post was a while ago?

What do you think he is, a paid blogger?

That old Law aphorism clearly applies here: If you have the law on your side, argue the law; if you have the facts, argue the facts; if you have neither, pound the table.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

Calendar

October 2018

M	T	W	T	F	S	S
1	2	3	4	5	6	7

8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

« Jul

Links

- @ramcq
- Collabora
- Planet Collabora
- Planet GNOME

Archives

- October 2018
- July 2017
- May 2010
- October 2009
- August 2009
- July 2009
- March 2009
- January 2009
- July 2008
- June 2008
- April 2008
- May 2007
- January 2007
- December 2006
- June 2006
- April 2006
- March 2006
- November 2005
- October 2005
- September 2005
- August 2005
- July 2005
- May 2005
- April 2005
- March 2005

Meta

- Log in
- RSS
- Comments RSS
- Wordpress