LINUX

Developers are lazy, thus Flatpak

A look at Flatpak madness

MARTIJN BRAAM
2023-06-03 15:58:47.614000

In the last decade I have seen a very slow but steady shift to solutions for packaging software that try to isolate the software from host systems to supposedly make things

easier. My first experience with this was Docker, now Flatpak is the thing for desktop applications.

The promise of Flatpak

So the thing Flatpak is supposed to fix for me as developer is that I don't need to care about distributions anymore. I can bolt on whatever dependencies I want to my app and it's dealt with. I also don't need to worry about having software in distributions, if it's in Flatpak it's everywhere. Flatpak gives me that unified base to work on and everything will be perfect. World hunger will be solved. Finally peace on earth.

Sadly there's reality. The reality is to get away from the evil distributions the Flatpak creators have made... another distribution. It is not a particularly good distribution, it doesn't have a decent package manager. It doesn't have a system that makes it easy to do packaging. The developer interface is painfully shoehorned into Github workflows and it adds all the downsides of containerisation.

Flatpak is a distribution

While the developers like to pretend real hard that Flatpak is not a distribution, it's still suspiciously close to one. It lacks a kernel and a few services and it lacks the standard Linux base directory specification but it's still a distribution you need to target. Instead of providing seperate packages with a package manager it provides a runtime that comes with a bunch of dependencies. Conveniently it also provides multiple runtimes to make sure there's not actually a single base to work on. Because sometimes you need Gnome libraries, sometimes you need KDE libraries. Since there's no package manager those will be in seperate runtimes.

While Flatpak breaks most expectations of a distribution it's still a collection of software and libraries build together to make a system to run software in, thus it's a distribution. A really weird one.

No built in package manager

If you need a dependency that's not in the runtime there's no package manager to pull in that dependency. The solution is to also package the dependencies you need yourself and let the flatpak tooling build this into the flatpak of your application. So now instead of

being the developer for your application you're also the maintainer of all the dependencies in this semi-distribution you're shipping under the disguise of an application. And one thing is for sure, I don't trust application developers to maintain dependencies.

This gets really nuts by looking at some software that deals with multimedia. Lets look at the Audacity flatpak. It builds as dependency:

- wxwidgets
- ffmpeg
- sqlite
- chrpath
- portaudio
- portmidi

So lets look at how well dependencies are managed here. Since we're now almost exactly half a year into 2023 I'll look at the updates for the last 6 months and compare it to the same dependencies in Alpine Linux.

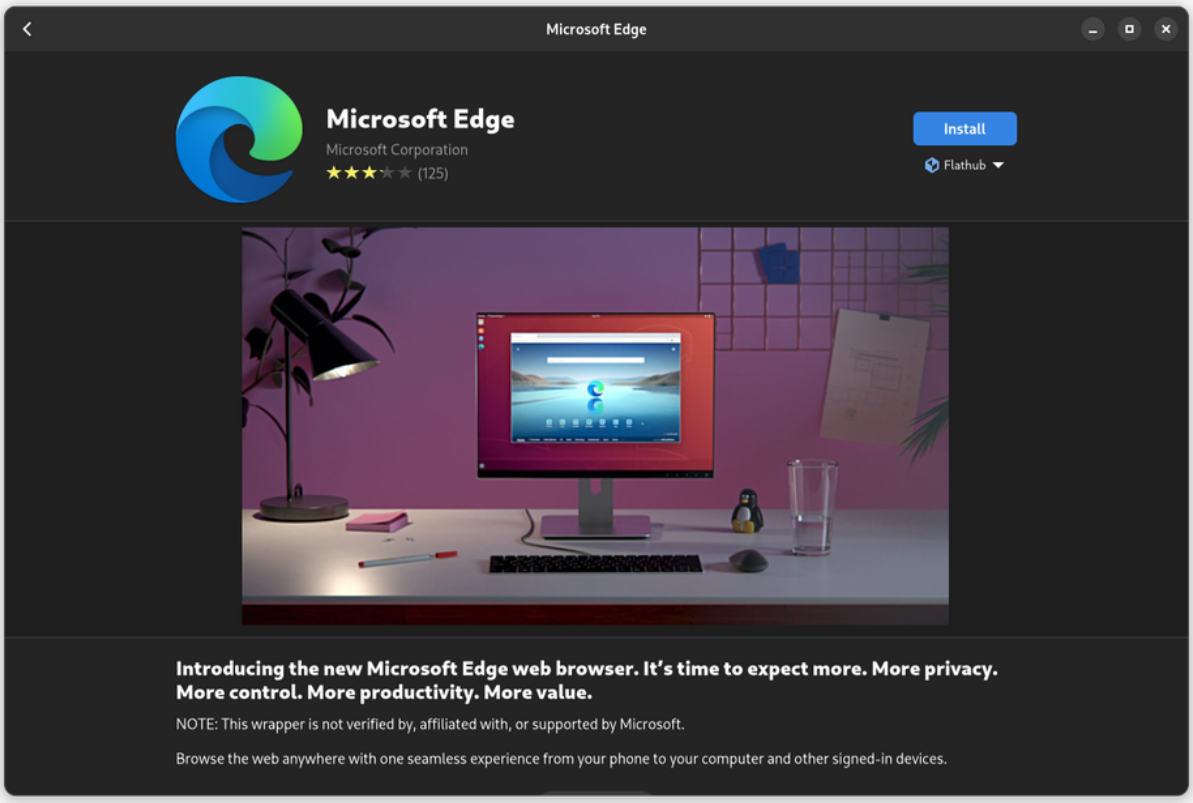
- audacity has been updated 4 times in the flatpak. It has been updated 5 times on Alpine.
- ffmpeg has been updated to 6.0 in both the flatpak and Alpine, but the ffmpeg package has had 9 updates because if codecs that have been updated.
- sqlite hasn't been updated in the flatpak and has been updated 4 times in Alpine
- wxwidgets hasn't been updated in the flatpak and has been updated 2 times in Alpine
- chrpath hasn't had updates
- portaudio hasn't had updates in flatpak and Alpine.
- portmidi hasn't had updates

This is just a random package I picked and it already had a lot more maintainance of the dependencies than the flatpak has. It most likely doesn't scale to have all developers keep track of all the dependencies of all their software.

The idea of isolation

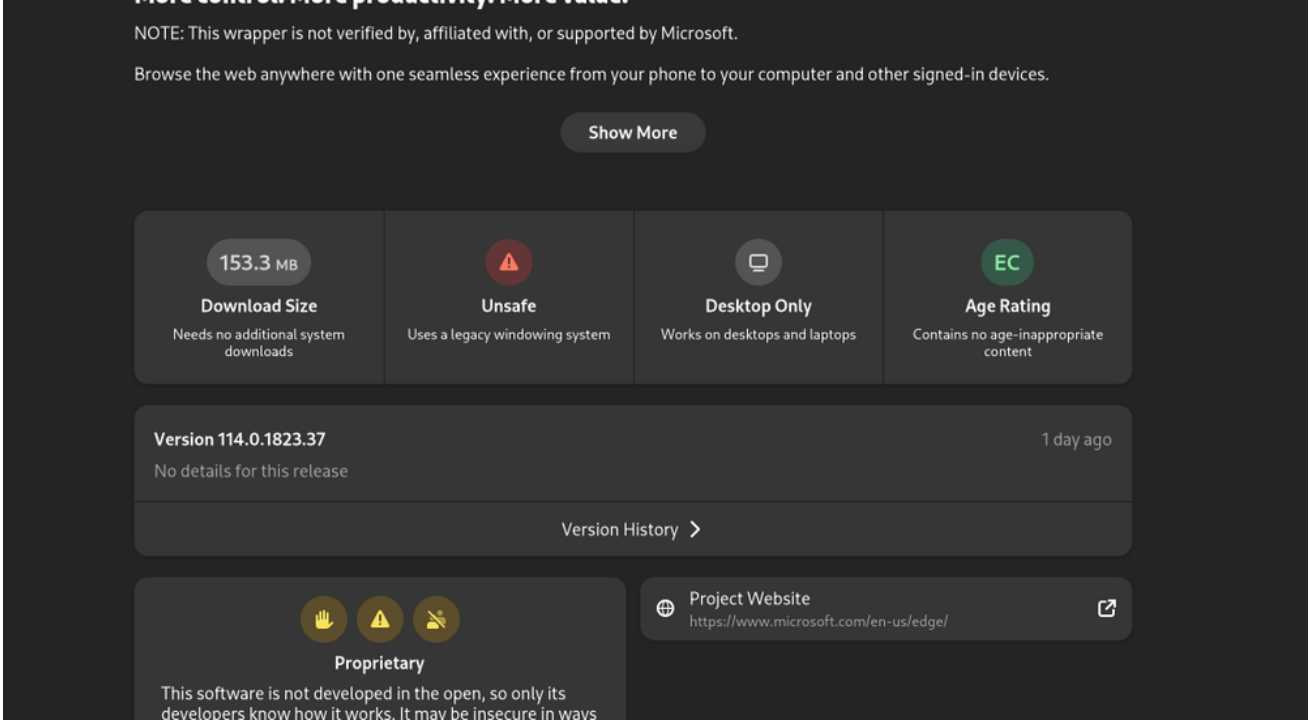
One of the big pros that's always mentioned with Flatpak is that the applications run in a sandbox. The idea is that this sandbox will shield you from all the evil applications can do so it's totally safe to trust random developers to push random Flatpaks. First of all this sandbox has the same issue any permission system that exists also has. It needs to tell the user about the specific holes that have been poked in the sandbox to make the application work in a way that end users *understand* what the security implications of those permissions are.

For example here's Gnome Software ready to install the flatpak for Edge:

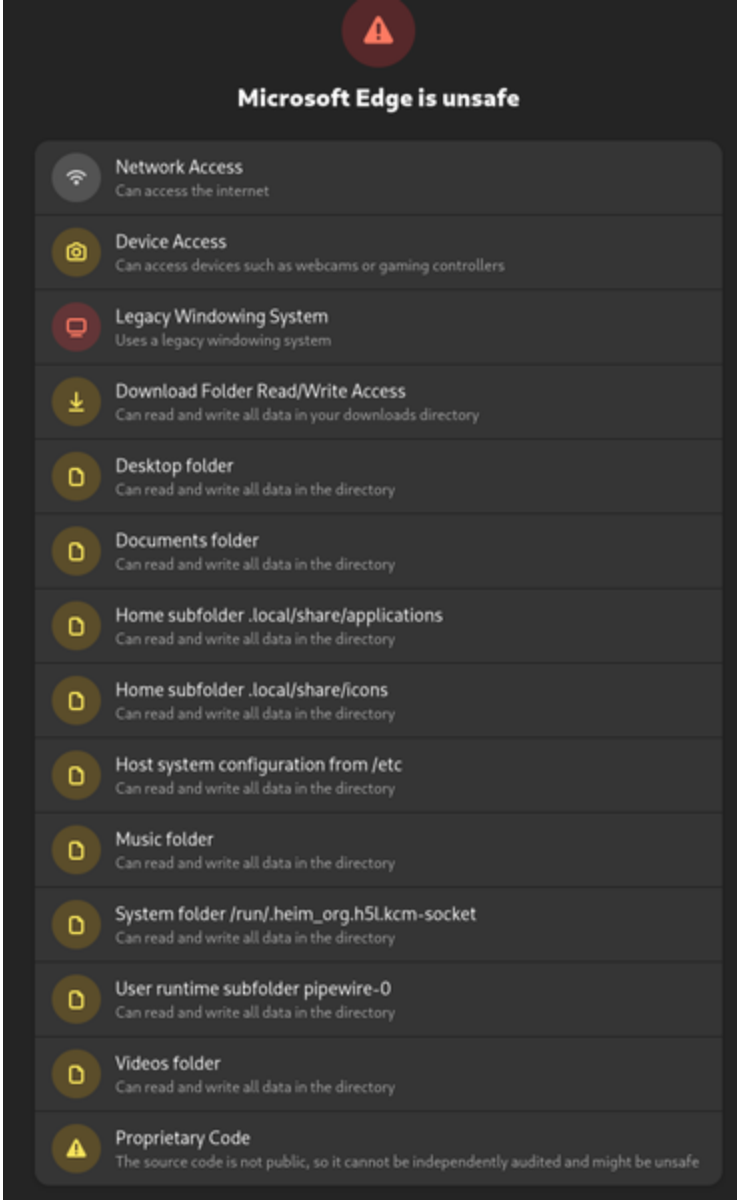


I find the permission handleing implemented here very interesting. There's absolutely no warning whatsoever about the bypassed security in this Flatpak untill you scroll down. The install button will immediately install it without warning about all the bypassed sandboxing features.

So if you *do scroll down there's more details right? Sure there is!*



There's a nice red triangle with the words Unsafe! pfew, everyone is fine now. So this uses a legacy windowing system which probably means it uses X11 which is not secure and breaks the sandbox. Well if that's the only security issue then it *might* be acceptable? Let's click that button.



Well yeah... let's hide that from users. Of course the browser needs to write to /etc. This is all unimportant to end users.

The even worse news is that since this is proprietary software it's not really possible to audit what this would do, and even if it's audited it's ridiculously easy to push a new more evil version to Flathub since practically only the first version of the app you push is thorougly looked at by the Flathub maintainers.

Even if there weren't so many holes in the sandbox. This does not stop applications from doing more evil things that are not directly related to filesystem and daemon access. You want analytics on your users? Just requirest the internet permission and send off all the tracking data you want.

So what about traditional distributions

I've heard many argument for Flatpaks by users and developers but in the end I can't really say the pros outweigh the cons.

I think it's very important that developers do not have the permissions to push whatever code they want to everyone under the disguise of a secure system. And that's *my opinion as a software developer*.

Software packaged by distributions has at least some degree of scrutiny and it often results in at least making sure build flags are set to disable user tracking and such features.

I also believe software in general is better if it's made with the expectation that it will run outside of Flatpak. It's not that hard to make sure you don't depend on bleeding edge versions of libraries while that's not needed. It's not that hard to have optional dependencies in software. It's not that hard to actually follow XDG specifications instead of hardcoding paths.

But packaging for distributions is hard

That's the best thing! Developers are not supposed to be the ones packaging software so it's not hard at all. It's not your task to get your software in all the distributions, if your software is useful to people it tends to get pulled in. I have software that's packaged in Alpine Linux, ALT Linux, Archlinux AUR, Debian, Devuan, Fedora, Gentoo, Kali, LiGurOS, Nix, OpenMandriva, postmarketOS, Raspbian, Rosa, Trisquel, Ubuntu and Void. I did not have to package most of this.

The most I notice from other distributions packaging my software is patches from maintainers that improve the software, usually in dealing with some edge case I forgot with a hardcoded path somewhere.

The most time I've ever spent on distribution packaging is actually the few pieces of software I've managed to push to Flathub. Dealing with differences between distributions is easy, dealing with differences between runing inside and outside Flatpak is hard.

But Flatpaks are easier for end users

I've ran into enough issues as end user of flatpaks. A package being on Flathub does not mean that it will be installable for an end user. I've ran into this by installing packages on the PineBook Pro which generated some rather confusing error messages about the repositories missing. It turns out that the Aarch64 architecture was missing for those flatpaks so the software was just not available. Linux distributions generally try to enable as much architectures as possible when packaging, not just x86_64.

A second issue I've had on my Pinebook Pro is that it has a 64GB rootfs. Using too many flatpaks is just very wasteful of space. In theory you have a runtime that has your major dependencies and then a few Megabytes of stuff in your application flatpak. In practice I nearly have an unique platform per flatpak installed because the flatpaks depend on different versions of that platform or just on different platforms.

Another issue is with end users of some of my Flatpaks. Flatpak does not deal well with software that communicates with actual hardware. A bunch of my software uses libusb to communicate with sepecific devices as a replacement for some Windows applications and Android apps I would otherwise need. The issue end users will run in to is that they first need to install the udev rules in their distribution to make sure Flatpak can access those USB devices. For the distribution packaged version of my software it Just Works(tm)

Flatpak does have it's uses

I wouldn't say Flatpak is completely useless. For certain usecases it is great to have available. It think Flatpak makes most sense for when closed source software would need to be distributed.

I would like to see this be more strict though. I wouldn't want to have flatpaks with holes in the sandbox with a proprietary license for example. Which is exactly what the Edge flatpak is.

It's quite sad that Flatpak madness has gone so deep into the Gnome ecosystem that it's now impossible to run the nice Gnome Builder IDE without having your application in a flatpak. (EDIT: Turns out that using Builder without Flatpak is possible again)

I don't think having every app on a Linux machine being Flatpak is anything I'd want, If I wanted to give developers that much power to push updates to anywhere in my system

without accountability I'd just go run Windows.

Latest Posts