



# Flatpak Security and Development

By TechScribe

---

## LICENSING

---

**TechScribe** is the author of this documentation and is the sole copyright owner of this documentation apart from book cover photo and software used within this documentation.

The license of this documentation is granted to you under Creative Commons NonCommercial license 3.0 which you can find more information from <https://creativecommons.org/licenses/by-nc/3.0/legalcode> or by contacting [TechScribe@linuxdev.app](mailto:TechScribe@linuxdev.app).

The book cover photo is owned by Sufyan on Unsplash and the license for this photo is granted under Unsplash License found on <https://unsplash.com/license>.

It should be noted that this book is a work in progress and is incomplete. This is merely a small informal documentation on Flatpak specifically. More information will be added to this book as I progresses through my study and research of Flatpak tool.

# Table of Contents

Licensing . . . . .	ii
<b>1 What is Flatpak?</b>	<b>1</b>
1.1 Flatseal Tool . . . . .	2
1.2 Practical Demonstration . . . . .	3
1.3 Gaming with Flatpak . . . . .	3



# WHAT IS FLATPAK?

Flatpak is a desktop application distribution containerization technology, essentially it jails your application into its own little world through the security feature called Sandboxing.

Let's hypothetically say we have an editor software downloaded and installed on our machine. On its own it is fairly harmless, but if we add random extension from the marketplace that it may not have been audited by other developers, we are subjecting ourselves to increased risks of running malicious software and codes. It could for instance overwrite our `.bashrc` script which overrides default bash behavior to alias existing program such as `sudo` which can then be used to harvest/load malicious software to hijack our machine with ease.

You can install flatpak by running the following for your respective operating system with the handy table below or to refer to this webpage: <https://flatpak.org/setup/>.

Distribution	Command
ArchLinux	<code>pacman -S flatpak</code>
Debian/Ubuntu/PopOS	<code>apt install flatpak</code>
OpenSUSE/SUSE	<code>zypper install flatpak</code>
Red Hat Enterprise Linux/CentOS	<code>yum install flatpak</code>
Fedora	Already installed
NixOS	Please refer to <a href="https://flatpak.org/setup/NixOS">https://flatpak.org/setup/NixOS</a> for instructions.

You may want to enable flathub repository for installing flatpak application.

```
flatpak remote-add --if-not-exists flathub \
https://flathub.org/repo/flathub.flatpakrepo
```

## FLATSEAL TOOL

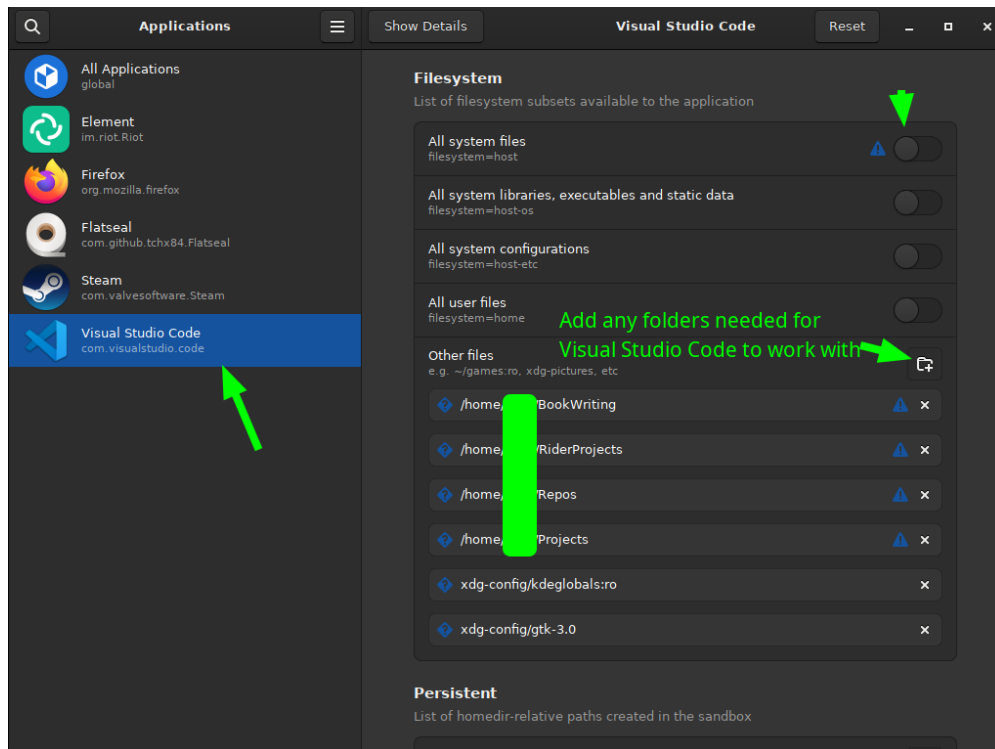
Flatseal Program is a frontend software that allows us to modify the permission of installed flatpak applications and we can limit or restrict what program is permitted to use. We can install flatseal by running the following command:

```
flatpak install flatseal
```

For this demonstration, we will install "Visual Studio Code" flatpak application and then open it with flatseal to view it's privileges.

```
flatpak install com.visualstudio.code  
flatpak run com.github.tchx84.Flatseal
```

The privilege "All system files" permission can be turned off after clicking on Visual Studio Code tab on the left side and you can selectively add specific folders that you want to allow Visual Studio Code to have control over:



---

## PRACTICAL DEMONSTRATION

---

Let's suppose we have the following malicious C# code that its sole intention is to alias sudo command to run a bad software in substitute.

```
1 Console.WriteLine("Now attempting to overwrite .bashrc!");
2 var currentDir = Environment.CurrentDirectory;
3 var nameOfDir = "";
4 while (!string.IsNullOrEmpty(currentDir) &&
5     currentDir != (Directory.GetParent(currentDir)?.FullName ?? string.Empty)
6     && nameOfDir != Environment.UserName.ToLower())
7     {
8         currentDir = Directory.GetParent(currentDir)?.FullName ?? string.Empty;
9         var idx = currentDir.LastIndexOf(Path.DirectorySeparatorChar);
10        idx = idx < 0 ? 0 : idx + 1;
11        nameOfDir = currentDir.Substring(idx);
12    }
13 var bashrcFilePath = Path.Join(currentDir, ".bashrc");
14 if (File.Exists(bashrcFilePath))
15     File.AppendAllText(bashrcFilePath, "\nalias sudo = badsoftware\n");
16 else
17     File.WriteAllText(bashrcFilePath, "\nalias sudo = badsoftware\n");
18 Console.WriteLine($"Successfully written bad bashrc file to {bashrcFilePath}");
```

Now we can go ahead and run the program by opening integrated terminal by clicking on Terminal drop down menu in Visual Studio Code and then click on New Terminal menu item and then enter the command into the terminal at the bottom as followed:

```
dotnet run
```

Now you'll find the .bashrc file created in your flatpak home directory after running the following command:

```
nano ~/.bashrc
```

You'll notice that it only contains the malicious code, but nothing else when you open .bashrc. That .bashrc will only affects whatever is **INSIDE** that flatpak application and you won't be affected by it anytime you open your terminal program outside of flatpak package and if you open .bashrc file, you'll notice that it have yet to be modified by the malicious demonstration code above.

That's the power of Flatpak!

---

## GAMING WITH FLATPAK

---

When installing Steam with flatpak, you gains the same level of sandboxing and so anytime you install a moddable games such as RimWorld for instance, the mods running on RimWorld would not be able to hijack your computer.

