# Open Source

Zach K

# Introduction

- Hello, I'm Zach, effectively a Junior - www.mathgeniuszach.com
- Big fan of open source - https://github.com/xMGZx
- Open source is great!
  - Linux is great!
- However, it's not right for everything.
  - My website is not open source, though pieces of it are.
- A lot of people don't understand open source!
  - Many companies feign "open source" for their own benefit.
- That's what this talk is for!

# What is Open Source?

- Open source means the "source code" is open to the public.
  - For programs, you can visibly see how the program was built.
  - You can also audit the code to check for any bugs - more on this later
  - Sometimes "source code" refers to more than just computer code.
- Being "open" typically means more than you might think.
  - Often to clarify this definition, the terms Free and Open Source Software (FOSS) or Libre (Free) are often used. (1)
  - The free here describes "free as in freedom", rather than "free as in free beer". (2)
    - Freedom means you can make changes and distribute them freely.
    - Some companies like to avoid giving you this freedom. They are not FOSS.
- For this talk, I'll stick to a more general definition of open source.

# Open Source and Other Licenses

- The easiest way to understand open source is by looking at common licenses.
- The simplest form of open source, is public domain. (Public Domain, CC0)
- Permissive licenses grant similar usage rights, but typically require attribution and license inclusion. (MIT, Apache, CC BY)
- Copyleft licenses grant usage rights, but forbid proprietization. (GPLv2, GPLv3, CC BY-SA)
- Some licenses include a "non-commercial" clause to prevent commercial usage. (JRL, CC BY-NC, CC BY-NC-SA)
- All other content either falls into proprietary licenses (copyright, default), or
- Trade secrets (usually protected by NDAs).
- Open source licenses are often necessary to prevent misuse.
- If you want to open source license your project (code or not), https://choosealicense.com/ is a good place to find the perfect license.

# Examples of Open Source

- VLC - A FOSS cross-platform multimedia player.
- GIMP - A FOSS cross-platform image editor.
- Python - The programming language.
- VSCodium - A FOSS code editor based on the VSCode editor. Removes telemetry and the link to the Microsoft Azure-based extension store, though that extension store can be added back.
- Firefox - An open source web browser (which has a lot of terrible defaults).
- Linux (or GNU/Linux) - Originally a kernel, refers to an entire family of operating systems based on the original kernel.

# Sketchy Open Source

- Google Chrome is not open source. Chromium is.
- Unreal Engine requires you to agree to a EULA to use their engine. After that, you can use the engine's source code with a bit of restrictions.
- John Deere (farming equipment manufacturers) use modified GPL licensed software but they only let you get it with a CD. The Software Freedom Conservancy is having trouble doing that though... (3)
- This isn't a complete list...

# The History of Open Source

- Sharing information openly for everyone to benefit is not a new concept. Neither is witholding it to benefit yourself.
- The standard scientific method includes publishing your report.
- In the 1950s and 1960s, software, like the computer, was mostly produced by hardware creators, academics and researchers under an open source mindset. (4)
- Late 1960s and 1970s, this started to change.
  - The growing software industry led to *United States vs IBM* in 1969, an anti-trust lawsuit ruling in part that their open bundled software was anticompetitive (5). This was, of course, withdrawn in 1982. (6)
  - In 1974 CONTU's decided computer programs ought to be copyrightable, and in *Apple v. Franklin* in 1983 source code gained the copyright status of a literary work (7).
  - Software licenses were growing more popular with companies like Microsoft and Apple forming and creating home computers in the late 1970s and 1980s.

# The History of Open Source (cont.)

- Open source was not dead though.
  - Scattered throughout the internet were random forums from people sharing their code.
- One person who really hated the trend of where software was going was Richard Stallman. In contrast to it, he launched the GNU (GNU's Not Unix!) project in 1983, which included the GNU GPL (General Public License). (9)
- In 1991, Linus Torvalds released the Linux kernel, eventually under the GPL in 1992. (10)
- In 2005, Linus Torvalds created Git as a FOSS DVCS to unify work on Linux (11). Notably from this many hosting sites like GitLab and GitHub have been created to make Git development significantly easier.

# Benefits of Open Source

- Program Transparency and Privacy
  - A key aspect of good open source code is that it is inherently designed to be fully transparent to the end user on what it does. These programs are "open" to be visibly checked and read by everyone.
  - There is a temptation from closed-source developers (typically out of profit), to perform survallance and other malware-like actions on their end users because they can't change the source code or notice it occuring.
- Flexibility
  - Open source code ought to be easily modifiable, and therefore is often able to be changed to work or be improved in a variety of circumstances.

# Benefits of Open Source (cont.)

- Security
  - A common misunderstanding with open source code is that it lacks security because the source code is not obscured. This principle is called "security by obscurity", and is not fully accurate.
  - Sidenote, one of the biggest things that makes a system insecure is it's popularity. More targets means more potential profit for threat actors.
  - Open source code gains it's security through routine audits and patches done by the community, for the community. There are good actors, just as much as bad actors, that use the code to find and patch bugs too.
- Community
  - Good open source projects are built on a strong sense of community trust, backing, and goodwill. Not everyone works for money! People do things out of collective needs, and many are just generous with their spare time.
  - Big companies can sometimes come together against monopolies. The Overture Maps Foundation is a good example. (11)

# The Pitfalls of Open Source

- Lack of Reliable Funding
  - Because open source software is open, it's typically possible to gain the program for free.
  - Developers rely on other forms of income like donations, selling only the binary (Aseprite), or perhaps none at all. These are very unreliable - the core.js project is a great example.
  - Businesses that only care about their bottom line rarely give donations or even support. Many use it "as convienient and discard it when its not". (12)
- Competition
  - This is normally a good thing. When a project is open source, anyone is free to clone it and make changes in their own way and eventually become a competitor of the original project.
  - Userbases will move to whatever project they think is better, even though it may be proprietary instead of open source (copyleft licenses prevent this).
  - For business making a living, having your userbase stolen can be pretty bad.
- Lack of support/community
  - Many open source pieces of software, for one reason or another, may lose their community entirely. When this happens, the software can become poor or unusable. A project is not automatically perfect or better if it's open source.

# Conclusion

- Open source, in spirit, is not a new concept.
- Open source is not insecure simply because it is open. This is a misrepresentation of security.
- Open source licenses and the idea is necessary, because the default is copyright.
- Open source is not right for everything; it requires either goodwill or funding to function, and without funding it may fail.
- Open source is awesome (and some may even say beautiful) because it enables communities and individuals to come and work together to achieve something for everyone's benefit.

# Sources


1. Why Open Source Misses the Point (Richard Stallman) - https://www.gnu.org/philosophy/open-source-misses-the-point.html
2. The Free Software Definition (Free Software Foundation) - https://www.gnu.org/philosophy/free-sw.html.en
3. John Deere's ongoing GPL violations: What's next (Denver Gingerich) - https://sfconservancy.org/blog/2023/mar/16/john-deere-gpl-violations/
4. Free and Open Source Software and FRAND-based patent licenses (Catharina Maracke) - https://doi.org/10.1111/jwip.12114
5. United States' Memorandumon on the 1969 case - https://www.justice.gov/atr/case-document/united-states-memorandum-1969-case
6. *In re Intern.* Business Machines Corp., 687 F.2d 591 (2d Cir. 1982)
7. *Williams Electronics, Inc. v. Artic Intl, Inc.*, 685 F.2d 870 (1982)
8. The GNU Project (Richard Stallman) - https://www.gnu.org/gnu/thegnuproject.html
9. Release Notes for Linux v0.12 (Linus Torvalds) - https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.12
10. Re: Trivia: When did git self-host? (Linus Torvalds) - https://marc.info/?l=git&m=117254154130732
11. Joint Development Foundation Announces Overture Maps Foundation to Build Interoperable Open Map Data (The Linux Foundation) - https://www.linuxfoundation.org/press/linux-foundation-announces-overture-maps-foundation-to-build-interoperable-open-map-data
12. The Problems of Open Source (Dr. Mark Tarver) - https://marktarver.com/problems.html