# Copy left and Open source

V.C.VIVEKANANDAN
MHRD IP chair Professor
NALSAR University of Law

# ORIGINS OF COPYLEFT

- RICHARD STALLMAN A MIT PROFESSOR STARTED THE GNU PROJECT in 1983 ( thirty years ago TO COUNTER THE PROPRIETORY RESTRICTIONS OF SOFTWARE COMPANIES
- “GNU's Not Unix!”. "GNU" is pronounced *g'noo, as one syllable, like saying "grew" but replacing the r* with *n*

- *SEE www.gnu.org for complete details*

# FREE SOFTWARE FOUNDATION (FSF)

- STALLMAN WENT ON TO START THE FREE SOFTWARE FOUNDATION IN 1984

- DON HOPKINS – COLLEAGUE OF STALLMAN WHO INTRODUCED COMPUTER GAMES LIKE SIM CITY AND THE SIMS USED THE WORD COPYLEFT- ‘ ALL RIGHTS ARE REVERSED’

# BIRTH OF LINUX

- In 1991, LINUS Torvalds a Finnish student unveiled a hobby he had been working on in a USENET newsgroup, an operating system kernel based on the Minix OS previously invented by Andrew Tanenbaum.
- This OS named as Linux or Linus Minix caught on as an alternate to the proprietary software as there is a freedom  to develop it the way one wants

# COPYLEFT CLARIFIED

- Current Linux operations has only 2 % of what Linus Torvalds developed
- With the advent of internet the copy left concept got a greater impetus
- However copy left as a concept is not the opposite of copyright but rather used as a model where it does not restrict the user to modify the software for further uses and also cannot place restrictions on what they have developed

# COPY LEFT AND COMMERCIAL USE

- The free  in free software refers to what users are allowed to do with it, which is whatever they want, provided they follow Copy left guidelines.

- It does not indicate that something is free of charge. In fact, the FSF has an initial distribution price for its GNU OS.

- And even if it is no cost, like Linux, there are other ways to make money off of it, as companies like Red Hat and IBM have found.

- Even Microsoft employs open-source development techniques in-house, though once its products hit the shelves, you would better have a license if you want to use any of them.

# THE FOUR RULES OF FREE SOFTWARE (FSF)

- A program is free software if the program's users have the four essential freedoms:
- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

# WHAT IS OPEN SOURCE?

- The term "free software" is older, and is reflected in the name of the free software foundation (FSF) founded by Stallman. The term “Open Source” came into existence in 1998 when a group wanted to promote and distribute software as a practical process than an ideological platform of Stallman

# The Ten Point agenda of Open Source Software (OSS)

- 1. Free Redistribution
- The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
- .

- 2. Source Code
- The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

- 3. Derived Works
- The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software

- 4. Integrity of The Author's Source Code
- The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

- 5. No Discrimination Against Persons or Groups
- The license must not discriminate against any person or group of persons.
- .

- 6. No Discrimination Against Fields of Endeavor
- The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research

- 7. Distribution of License
- The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

- 8. License Must Not Be Specific to a Product
- The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

- 9. License Must Not Restrict Other Software
- The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

- 10. License Must Be Technology-Neutral
- No provision of the license may be predicated on any individual technology or style of interface.

# Licensing

- At its core: The verb **license** or **grant license** means to give permission

- A licensor may grant a **license** under intellectual property laws to **authorize a use** (such as copying software or using a patented invention) to a licensee, sparing the licensee from a claim of infringement brought by the licensor.

# Open Source Software

From a legal perspective, the *availability* of the source code for OSS and the right to modify and improve the code is an important distinction between OSS and commercial software---but in both cases, restrictions on further use exist.

# The Licenses

- All Open Source Software licenses are NOT the same

- Each license carries with it a different set of requirements for using the software and outlines a different set of requirements for modifying the source code…among other terms

# Copyleft vs. Non-Copyleft

- A major difference between Open Source licenses is whether the license is considered “copyleft” or not

- Where copyright law allows the copyright owner to withhold permission to copy, modify, or distribute software, Copyleft licenses **require** that permission to be granted

# Copyleft Cont.

- Copy left licenses are conditional licenses.  In order to use or distribute software licensed under a copy left license any changes you make to the software must be released under the same license.

- A copy left license makes sure that all modified versions of the software remain free and open in the same way the original software was

- The GPL is considered the most popular Copy left License

# Permissive Licenses

- Permissive Licenses, like the BSD License, are non-copyleft licenses
- Permissive Licenses do not place many restrictions on later development or modification of the original software
- Since Permissive Licenses do not place heavy restrictions on subsequent use, they do not preserve software rights in downstream versions
- If your software is licensed under a permissive license, subsequent developers can use your permissively licensed code in closed source proprietary software.

# Weak Copyleft – the LGPL

- The Lesser-General Public License (LGPL) is like the GPL, but it allows works licensed under it to be linked to by closed-sourced proprietary software; which would not be allowed under the GPL

- The LGPL was originally used for libraries. The LGPL would allow a developer to use a library of code licensed under the LGPL, without requiring the developer to release their software as open source. If the library itself is changed though, the copyleft provisions apply to the new version of the library.

# Important Questions to Ask

- When trying to figure out what license to use for your source code, there are a few basic questions you should ask
  - If the program is modified, can the results be distributed under a different license?
  - What are the risks of combining the program with proprietary software?
  - What other requirements are imposed by the license?

# Modifying the Program

- Copyleft licenses require any distribution of a modification to be distributed under the same license. Some copyleft licenses will allow distribution under a similar license

- The copyleft provisions do not kick in until distribution though. So you can typically modify a program and use the modified version internally without revealing the changes and thereby protecting any other code that could be considered a trade secret. **Once the modified version is distributed, the changes must be revealed**

# Combining OSS with Proprietary Software

- Copyleft software cannot be combined with proprietary software
- Non-copyleft software and some weak copyleft software (LGPL) can be combined with proprietary software

# Other Requirements

- Every license has additional requirements that must be complied with, some examples are a requirement to:
  - Include warranty disclaimers
  - Include copyright and attribution notices
  - provide a copy of the license to a downstream licensee
  - include a description of any changes made to the code by the licensee prior to redistribution
  - include an offer to provide the source code to the software upon request
  - include source code to non-standard software that is required in order for the program to run properly
  - include a file listing any known intellectual property disputes involving the software

# An Overview of the various licenses

| | BSD | Apache | GPLv2 | LGPL | GPLv3 |
|---|---|---|---|---|---|
| Copyleft? | No | No | Strong | Weak | Strong |
| Distribute Object Code without providing source code? | Yes | Yes | No | No | No |
| Distribute Derivatives Under a different License? | Yes | Yes | No | No | No |
| Copyright Notice Required? | Yes | Yes | Yes | Yes | Yes |
| Disclaimers Required? | Yes | Yes | Yes | Yes | Yes |
| Copy of License Required? | No | Yes | Yes | Yes | Yes |
| Notice of Changes Required? | No | Yes | Yes | Yes | Yes |
| Legal Information Required? | No | No | No | No | No |

# Consequences of Breaking a License

- In recent years there has been more litigation surrounding OSS.
- The consequences of breaking an open source license can be dire.
- Care must be taken both when using OSS, when distributing modifications of OSS, and when basing new software on OSS.

# BusyBox Litigation

- A series of lawsuits have been filed against companies for violating the GPL in relation to BusyBox software (released under GPL v.2)

- Multiple companies had been accused of using the BusyBox software without complying to the GPL requirements.
- Every case was settled out of court

- None of these suits involved modifications to the BusyBox source code. All suits were brought for violating requirements of the GPL, such as the requirement to include the source code, or the requirement to include attribution.

- **TAKE AWAY: Even if you don't change the program's source code you can still get into trouble!**

# Jacobsen v. Katzer, 535 F.3d 1373

- The most important case to date involving OSS is the Jacobsen Case
- Appellate court ruled in its August 13, 2008 decision that an open source licensor may pursue a claim for copyright infringement if the license clearly sets out conditions on the use of the software

# Jacobsen, The basic facts

- Robert Jacobsen, the plaintiff and a model train hobbyist, holds a copyright to software code that he makes available to the public free of charge under an open source license called the Artistic License.
- The defendants, Matthew Katzer and Kamind Associates, develop commercial software products for the model train industry and hobbyists using parts of Jacobsen's code.
- Jacobsen brought an action for copyright infringement and moved for a **preliminary injunction** against the defendants, accusing them of copying certain portions of his software code and incorporating it into their own commercially available software products without abiding by the terms of the Artistic License.

# Jacobson, continued

- In their defense, Katzer and the other defendants argued that the violation of the terms of the license agreement were merely violations of the CONTRACT, and not any copyrights.

- And…since this was just a violation of a CONTRACT, a preliminary injunction remedy is not applicable, nor do they have to pay out any $$$ since there are no actual damages because they breached an open source contract for which no money was exchanged.

# Jacobsen, Cont.

- Issue before the court:

Does failing to adhere to an open source license constitute Breach of Contract or Copyright Infringement?

# Jacobsen, Cont.

- District court sided with the Defendants and denied Jacobsen’s motion for a preliminary injunction.

# Jacobson Appeals

- Jacobsen appeals the denial of the preliminary injunction.
- This time, court sides with him.

# Appellate Court's Reasoning

- Generally, "a copyright owner who grants a non-exclusive license to use his copyrighted material waives his right to sue the licensee for copyright infringement" and can only sue for breach of contract

- But, if a license is limited in scope and the licensee acts outside the scope, the licensor can bring an action for copyright infringement.

- The Jacobsen decision found that if an open source license is limited in scope, then a licensee acting outside the scope would constitute a breach of the license, and would allow a copyright infringement suit to be brought.

# Jacobsen, Cont.

- The court used factors such as the express language of the license (“The intent of this document is to state the *conditions*”) and the traditional language used (“provided that”) in its analysis.

- The court found that “the restrictions were both clear and necessary to accomplish the objectives of the open source licensing collaboration, *including economic benefit*,” and were “vital to enable the copyright holder to retain the ability to benefit from the work of downstream users.”

# Status of Case Now

- Court vacated and remanded so the district court had to go back and figure out if, in light of the copyright infringement, a preliminary injunction WAS appropriate. (i.e. is their irreparable harm justifying the injunction---)

- District court denied the preliminary injunction the second time around, and gave Jacobsen leave to file an amended complaint. He did. An answer was filed and includes a counterclaim for copyright infringement!

# The Effect of Jacobsen on New OSS Licensing Litigation

- This ruling opens the door for other OSS licensors who are not directly profiting from the licensing of the copyrighted work to seek protection for their open source software.

# Emerging Litigation

- Earlier this year, The Free Software Foundation filed a complaint against Cisco alleging that Cisco violated both the GPL and LGPL licenses accompanying various GNU programs for which FSF owns copyrights, and that, as a result, Cisco has infringed on FSF-owned copyrights.

# FSF v. Cisco
# Under the *Jacobsen* Standard

- As a preliminary matter, a plaintiff needs to make out a prima facie case of copyright infringement.
  - In *Jacobsen*, the court cited the fact that
    - 1) the parties did not dispute that Jacobsen was the owner of valid copyrights, and
    - 2) that Katzen admitted to copying, modifying, and distributing part of Jacobsen's copyrighted work.

# FSF v. Cisco
# Under the *Jacobsen* Standard

- The same factors will likely be found in the FSF v. Cisco case. Assuming the FSF's complaint is accurate on the facts:
  - FSF owns copyrights for GNU C Library, CNU Coreutils, GNU Readline, GNU Parted, GNU Wget, GNU Compiler Collection, GNU Ginutils, and GNU Debugger
  - Cisco admitted on July 5, 2006, that it had distributed the model WIP300 and its Firmware without providing the corresponding source code as is required by the license.

# FSF v. Cisco
# Under the *Jacobsen* Standard

- In *Jacobsen*, once the prima facie case for copyright infringement was made, the court then turned to evaluate whether the scope of the license was limited in some way and whether the use of Jacobsen's copyrighted products by Katzen was outside the scope of the license.

# FSF v. Cisco
# Under the *Jacobsen* Standard

- The FSF complaint asserts that both the GPL and LGPL contain *conditions* on a licensee's use.
- However, unlike the license in *Jacobsen,* neither the GPL nor LGPL state explicitly language similar to Jacobsen's "the intent of this document is to state the conditions under which a Package may be copied." (practice note to lawyers…)

# FSF v. Cisco
# Under the *Jacobsen* Standard

- Both licenses use the "traditional language" identified in *Jacobsen* as creating conditions
  - "If...then" statements
  - "Provided that" clause
- Both licenses include the statement that "any attempt otherwise to copy, modify, sublicense, or distribute the Program [or link to the Library] is void, and will automatically terminate your rights under this License."
- The court in *Jacobsen* applied California law to determine whether "provided that" actually created a condition.

# Your License Under the *Jacobsen* Standard

- Does your license use scope-limiting terms?
  - Does the license employ natural conditional language?
  - Does your license include a portion explicitly stating its purpose in conditioning the use of the OSS?
  - What language creates a condition under each party's state's property law?

# Your License Under the *Jacobsen* Standard, Cont.

- Limiting Terms, Cont.
  - What language creates *only a covenant (as opposed to a "Condition")* in each party's state?
  - Does the license explain the effect of a licensee acting outside of the scope of the license?
- Can the licensor demonstrate an economic benefit, even indirectly, in the copyrighted OSS software?

# Damages

- Since failing to adhere to an open source license can be considered a copyright violation, the party breaching the license can be subject to multiple punishments including:
  - An injunction to stop the distribution and use of the software
  - Monetary Damages
  - Statutory Damages
  - Attorneys’ Fees

# Monetary Damages

- Disgorgement of Profits is usually the test used for monetary damages.
- NOTE: Valuing actual monetary damages associated with open source software is still a bit murkey
- For example: monetary damage for failing to attribute? Failing to post source code easily available elsewhere?
- (Katzer decision noted that during oral arguments, both parties agreed it would be difficult to ascertain monetary damages under *contract* law.)

# Statutory Damages

- Where actual damages are difficult to calculate, statutory damages might kick in.
- Statutory damages for copyright infringement in the United States can range from $750 to $30,000 “as the court considers just,” and in cases of willful infringement the maximum statutory damages are increased to $150,000.
- They also include….Attorneys’ fees which, depending on the firm and the case, can easily exceed the 150K maximum!

# Damages Takeaway

- Better safe than sorry otherwise:
  - 300K plus in legal fees if the case goes all the way to trial (just for your costs even if you win)
  - Injunctions
  - Actual damages
  - Statutory Damages when actual damages are tough to ascertain

# How to be Safe?
# Corporate Precautions

- Conduct an audit of software used and distributed by the company for the inclusion of OSS.
- Create a database of OSS used and distributed by the company. This database should include the results of the audit and should be updated regularly.
- Employ a software development management and version control system.
- Educate employees about the benefits and risks of using OSS and distribute periodic memoranda to keep employees updated as to the latest risks and benefits of OSS and any changes to the corporate policy.
- Adopt and distribute a Corporate Open Source Acquisition and Use Policy.

- Adopt OSS acquisition and use approval procedures identifying personnel authorized to approve OSS use.
- Train employees at all levels on how to comply with the policy and procedures.
- Choose a Software Review Board consisting of members of the IT Department, the Product Development Team, the Legal Department, and the Executive Management to review the requests for OSS use.
- Adopt an OSS use approval method for the Review Board including guidelines for balancing the benefits and risk of using OSS.

# continued

- If necessary, isolate he Development Team and Development Environment from the IT Department to prevent OSS creep into proprietary software.
- Create a list of favorable open source licenses for developers to consider but remind them that approval of the use of OSS is still required. Keep the list current.
- Conduct periodic reviews to ensure employees comply with the corporate policy and procedures.

- Conduct annual reviews of the policy to ensure it remains in line with the corporate objectives, business model, and structure and that it is working properly.
- Ensure commercial software providers include representations, warranties, and indemnifications in their license agreements stating that the software supplied does not contain any OSS or that the OSS included in the licensed software is clearly identified.
- Purchase an OSS insurance policy, if justified, to minimize risk

# Finally….

- Involve your lawyer from the beginning---- not once you’ve been served with a complaint. ☺

# Questions?

- Saper Law Contact Info:

  Daliah Saper

  500 N Dearborn, Suite 1200

  Chicago, IL 60654

  312.527.4100

  www.saperlaw.com