» The Open Compliance Program

A Glimpse Into Recommended Practices in a FOSS Compliance Management Process

Part I of 2

By Ibrahim Haddad (Ph.D.), The Linux Foundation

A White Paper By The Linux Foundation
http://www.linuxfoundation.org

Paper Organization and Background

This paper is one in a series that discusses the topic of FOSS compliance and it is part of the free educational material that the Linux Foundation is making available under the Open Compliance Program. This paper highlights some of the recommended practices and various considerations when integrating FOSS in commercial products and it is divided into two parts:

- Part I of the paper covers recommended practices that map to the various steps within a FOSS compliance management end-to-end process.
- Part II of the paper will follow with emphasis on FOSS compliance considerations in relation to source code modifications, notices, distribution, software design, usage, linkages and code mixing. Furthermore, Part II will provide a discussion of recommended practices related to the various building blocks in a FOSS compliance program (For a discussion on elements of a successful FOSS compliance program please refer to: http://www.linuxfoundation.org/lp/page/sign-up-for-foss-compliance-whitepaper2).

In parallel to publishing free educational material, the Linux Foundation is also making available the "Open Compliance Program Self-Assessment Checklist", an extensive checklist of compliance practices found in industry-leading compliance programs (hence the title of this paper – A Glimpse). Companies can use the Self-Assessment Checklist as a confidential internal tool to assess their progress in implementing a rigorous compliance process and help them prioritize their process improvement efforts. We provide more information about this checklist and its availability at the end of this paper.

Introduction

In previous papers published by The Linux Foundation (available from http://www.linuxfoundation.org/publications), we discussed a sample FOSS compliance end-to-end management process. As a refresher, the process illustrated in Figure 1 includes the various steps a software component goes through before it is approved for inclusion in a product software stack. The process starts by identifying the various software components integrated into the product's build system and ends by compiling a list of resulting license obligations. In the following sub-sections, we provide some of the recommended practices related to processing compliance requests. The recommended practices map directly to the steps illustrated in the compliance process.

OPEN COMPLIANCE PROGRAM

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org

THE LINUX FOUNDATION

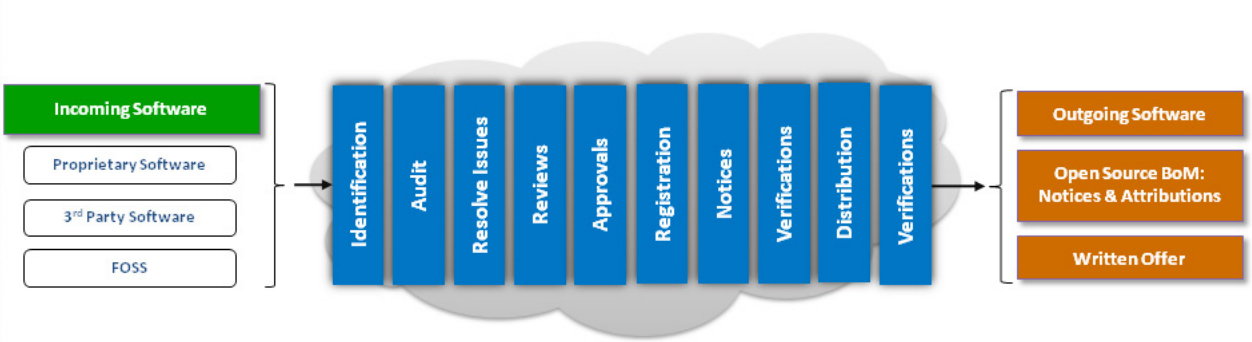

Figure 1. Illustration of a compliance end-to-end management process

Identification Phase

In the identification phase of a compliance process, organizations identify all of the components or elements entering their product's build system, their origin and license information. There are three main sources for incoming source code:

- Proprietary software, created by their own developers which may include snippets of FOSS or in many cases depends on or links to FOSS.
- Third party software, developed by third party software providers or consultants and made available under a commercial or FOSS license. This software category may include snippets of FOSS or in many cases depends on or links to FOSS.
- FOSS, developed by members of the FOSS community.

It is recommended to identify all the incoming software components and pass all of them into the source code auditing phase. based on tasks they receive from the core team or the OSRB.

Source Code Auditing

There are four recommended practices in relation to source code auditing or scanning:

- **Scan all source code:** It is recommended to scan every bit of source code incorporated into the product. Development teams may have introduced FOSS into proprietary or third party source code. Furthermore, development teams may have made modifications to FOSS triggering the need for additional due diligence and additional obligations to fulfill. Therefore, it is critical to identify and audit all source code included in a product.
- **Scan newer versions of previously approved packages:** There are instances where a previously approved package is either modified and used again (sometimes in a different context than the one it was approved for), or used as-is or with modifications in a different product, or even a new version is downloaded and applied in the product software stack. Since compliance is verified on a product-by-product basis, just because a FOSS package is approved for use in one product does not necessarily mean it will be approved for use in a second product. As a rule of thumb, each time developers modify a previously approved component or plan to use a previously approved component in a different product, the source code of the component should be re-scanned and the component should go

OPEN COMPLIANCE PROGRAM
1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org
THE LINUX FOUNDATION

through the approval process again.

- **Ensure that each new version of the same FOSS component is reviewed and approved:** When developers upgrade the version of a FOSS package, make sure that the license of the new version is the same as the license of the older used version. License changes can occur between version upgrades. Therefore it is recommended to go through the process again if the same component (older version) was approved in the past.
- **Scan early and often:** You might have heard of the "release early and often" open source development method which is an essential aspect of the FOSS development model whereby source code is released early and very often for users to experiment with and report bugs as part of ongoing quality assurance activities. "Scan early and often" follows the same spirit in encouraging of scanning source code early in the development process and continuing to do so regularly to ensure that compliance efforts are proceeding in parallel to the development efforts, and not lagging behind. Companies are also recommended to create a list of conditions that define when a new scan is required to make the process more efficient. "Scanning early and often" has several advantages:
  - It allows the discovery of compliance problems as they occur.
  - It allows providing solutions to discovered problem within acceptable timeframe.
  - It keeps the delta with the previous scan to a minimum.
  - It allows very efficient processing of incremental scans.

Resolving Issues

When the source code is scanned and compliance issues are flagged or discovered, there are a number of ways to resolve issues. Below are some guidelines for handling such cases:

- When in doubt with the scan results, discuss with Engineering (interview the developers responsible for the specific software component in question).
- Inspect and resolve each file or snippet flagged by the scanning tool as matching source code coming from sources other than where you think it came from.
- Identify any source code modifications to FOSS . Ideally, you should not rely on engineers to remember if they made code changes. You should rely on your build tools to be able to identify code changes, who made them and when.
- If the source code scanning tool identifies for instance the use of unauthorized GPL licensed source code within a proprietary component (that must remain proprietary), this will be reported to Engineering with a request for correction. It is highly recommended to re-scan the source code after Engineering has resolved the issue to get a confirmation that Engineering has removed the GPL source code and replaced it with proprietary source code.
- In preparation for the legal review, in addition to the source code audit report generated by the scanning tool, it is recommended to provide attorneys with all information discovered on the licensing of the specific component:
  - For FOSS components, this includes COPYING, README, or LICENSE files.

OPEN COMPLIANCE PROGRAM
1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org
THE LINUX FOUNDATION

- For software components received from a third party software provider, this includes the licensing agreement.

Reviews

There are different types of reviews that occur as part of a compliance process. In this section, we discuss two such reviews: Architecture review and linkage analysis review. The architecture review is an analysis of the interaction between FOSS and proprietary and third party software components. Companies often conduct the architecture review with the architect responsible of the product in question in addition to the developers responsible for the various key software components. The goal of this review is to identify:

- Components that are FOSS (used "as is" or modified)
- Components that are proprietary
- Components that are third party licensed under a commercial license
- Component dependencies
- Communication protocols
- Dynamic versus static linking (discussed in the following section)
- Components that live in kernel space versus user space
- Components that use shared header files
- Other FOSS that the specific software component interacts with or depends on especially if it is governed by a different FOSS license

The result of the architecture review is an analysis of the licensing obligations that may extend from the FOSS to the proprietary or third party components.

An extension to the architecture review is the linkage analysis review that aims to find potentially problematic code combinations at the static and dynamic link level. A recommended practice is to perform dynamic and static linkage analysis for every binary in the build system. The goal is to determine if any FOSS obligations are extending to proprietary or third party software components.

Such review and analysis is best conducted through automated tools that can crawl the build system and flag possible linkage conflicts based on predefined rules and policies. If a linkage issue was found, a bug ticket is assigned to Engineering with a description of the issue in addition to a proposal on how to solve the issue. Once Engineering confirms that the issue has been resolved, it is highly recommended to re-do the linkage analysis to verify that the code changes introduced by Engineering have in fact resolved the linkage issue without introducing new issues.

OPEN COMPLIANCE PROGRAM

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org

THE LINUX FOUNDATION

The Dependency Checker Tool

The Dependency Checker Tool was released by the Linux Foundation (under the MIT license) as a part of the Open Compliance Program to help companies and individuals with their compliance due diligence. Like any other open source project, the project is open (open mailing list, open git repository, open bugzilla) and governed by open source development processes. The goal of the tool is to flag problematic code combinations at the dynamic and static link level. The tool identifies a linkage conflict between binaries and libraries based on license policies defined in advance by the user of the tool . The tool provides a number of features and capabilities and can be used as a platform to innovate and enable easy compliance.

Resources

- Access source code via git: http://git.linuxfoundation.org/?p=dep-checker.git
- Subscribe to the mailing list: https://lists.linux-foundation.org/mailman/listinfo/dep-checker-dev
- File bugs or feature requests via bugzilla: http://bugs.linuxfoundation.org (Select "Compliance" project, then "Dependency Checker Tool" component)
- Download the Dependency Checker Tool paper: http://www.linuxfoundation.org/sites/main/files/publications/lf_foss_compliance_dct.pdf

Approvals

As part of the approval step in the compliance process, there are two main recommended practices:

- Verifying that all sub-tasks related to the compliance ticket have been completed and closed before approving the compliance ticket. Often it is easy to forget sub-tasks or pending sub-issues, which may led to prematuring closing a compliance ticket even though open issues remain.
- Recording a summary of the discussions that lead to the decision of approval or denial. It can be very useful when receiving a compliance inquiry to be able track on which basis approval was provided for a given component and track how issues were resolved.

Notices

Companies using FOSS in their products need to:

- Acknowledge the use of FOSS by providing full copyright and attribution notices.
  Inform the end user of their product how to obtain a copy of the FOSS source code (when applicable, for example in the case of GPL and LGPL).
- Reproduce the entire text of the license agreements for the FOSS code included in the product.

Some of the recommended practices in this area include:

- Collect of attribution an license notices as FOSS is approved for inclusion in the product.

OPEN COMPLIANCE PROGRAM

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org

THE LINUX FOUNDATION

Following this method, the notice file companies need to publish will always be up-to-date and will include lists of all FOSS, license information, copyright and attributions notices.
- Use clear language in the written offer and be inclusive of all FOSS included in the product.
- Ensure that the end users of the product know how to locate this information either on the product itself, in the product documentation (user manual or CDROM) and/or on a web site.

Verifications

Due to the large number of verification steps, it is essential to develop, maintain and evolve checklists that cover all the verification steps the compliance team follows to ensure consistency and to ensure that no verification steps are overlooked.

Examples of pre-distribution verifications include:

- Verification that all FOSS packages destined for distribution have been identified and approved.
- Verification that inappropriate comments have been removed from the source code packages.
- Verification that the source code packages (including modifications) have been verified to match the binary equivalence shipping in the product.

Verification that all appropriate notices have been included in the product documentation in addition to the availability of a written offer to inform end-users of their right to request source code for identified FOSS (when applicable).

Once FOSS packages are uploaded to the distribution web site, there are some recommended post distribution verification steps, including for instance:

- Verification that the package has been uploaded correctly.
- Verification that the packages can be downloaded and uncompressed on an external computer without any errors.
- Verification that the packages compile properly.

OPEN COMPLIANCE PROGRAM

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org

THE LINUX FOUNDATION

The Code Janitor Tool

Before releasing source code to the public, companies often perform a linguistic review to make sure that developers did not leave any comments about future products, product code names, mention of competitors, or any inappropriate comments. The Code Janitor tool was initiated by the Linux Foundation as an Open Source project and it allows its users to create a database of inappropriate words (or keywords) the tool will use in a brute force scan of the code. The result is a list of files that contain the offending "keywords."

Resources

- Access the source code via git: http://git.linuxfoundation.org/janitor.git
- File bugs or feature requests via bugzilla: http://bugs.linuxfoundation.org (Select "Compliance" project, then "Code Janitor Checker Tool" component)
- Subscribe to the mailing list: https://lists.linux-foundation.org/mailman/listinfo/code-janitor-dev
- Download the Code Janitor Tool overview paper: http://www.linuxfoundation.org/sites/main/files/publications/lf_foss_compliance_cjt.pdf

Conclusion

In this paper, we covered some of the recommended practices related to the various steps in a sample FOSS compliance end-to-end process. In the following paper, we will continue discussing this topic with emphasis on FOSS compliance considerations in relation to source code modifications, notices, distribution, software design, usage, linkages and source code mixing. Stay tuned for the second installment of this paper.

Linux Foundation Resources

- **Open Compliance Program:** http://www.linuxfoundation.org/programs/legal/compliance
- **Professional and Comprehensive Compliance Training:** The Linux Foundation offers hands-on training from compliance experts for individuals and companies responsible for achieving compliance with open source licenses and establishing an open source compliance program, as well as for those who simply want to learn more about compliance. Options available include live onsite training in addition to instructor-led live remote training. http://www.linuxfoundation.org/programs/legal/compliance/training-and-education
- **Compliance Publications:** http://www.linuxfoundation.org/publications
- **Open Compliance Directory and Rapid Alert System:** http://www.linuxfoundation.org/programs/legal/compliance/directory
- **Compliance Tools:** http://www.linuxfoundation.org/programs/legal/compliance/tools
- **The Software Package Data Exchange™:** The SPDX™ specification is a standard format for communicating the components, licenses and copyrights associated with a software

OPEN COMPLIANCE PROGRAM

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org

THE LINUX FOUNDATION

package. http://www.spdx.org/

- **FOSSBazaar:** An open community of technology and industry leaders who are collaborating to accelerate adoption of free and open source software in the enterprise. http://fossbazaar.org/

Coming Soon: Open Compliance Program Self-Assessment Checklist

The Linux Foundation is compiling an extensive checklist of practices found in industry-leading compliance programs. Companies can use this checklist as a confidential internal tool to assess their progress in implementing a rigorous compliance process and help them prioritize their process improvement efforts. The checklist will be available for download from the Linux Foundation web site on 11/01/2010. To request a copy, please visit: http://www.linuxfoundation.org/lp/page/self-assessment-checklist-inquiry

Acknowledgments

The author would like to express his gratitude to Karen Copenhaver (Legal Director of the Linux Foundation and Partner in Choate, Hall & Stewart LLP 's Business & Technology practice) and to Philip Koltun (The Linux Foundation Director of Open Compliance Program) for their reviews and valuable input.

About the Author

Dr. Ibrahim Haddad manages The Linux Foundation's Mobile Linux initiatives and works with the community to facilitate a vendor-neutral environment for advancing the Linux platform for next-generation mobile computing devices.

About the Open Compliance Program

The Linux Foundation's Open Compliance Program is the industry's only neutral, comprehensive software compliance initiative. By marshaling the resources of its members and leaders in the compliance community, the Linux Foundation brings together the individuals, companies and legal entities needed to expand the use of open source software while decreasing legal costs and FUD. The Open Compliance Program offers comprehensive training and informational materials, open source tools, an online community (FOSSBazaar), a best practices checklist, a rapid alert directory of company's compliance officers and a standard to help companies uniformly tag and report software used in their products. The Open Compliance Program is led by experts in the compliance industry and backed by such organizations as the Adobe, AMD, ARM Limited, Cisco Systems, Google, HP, IBM, Intel, Motorola, NEC, Novell, Samsung, Software Freedom Law Center, Sony Electronics and many more. More information can be found at http://www.linuxfoundation.org/programs/legal/compliance.

OPEN COMPLIANCE PROGRAM

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org

THE LINUX FOUNDATION

The Linux Foundation promotes, protects and advances Linux by providing unified resources and services needed for open source to successfully compete with closed platforms.

To learn more about The Linux Foundation, the Open Compliance Program or our other initiatives please visit us at http://www.linuxfoundation.org/.