Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 7, Number 1, pages 1–4, January 2018

KyAnalyzer: An Efficient Code Source Classifier for Operating System Packages

(preliminary version)

Yi Ren, Jianbo Guan, Hao Zhu, Liang Bai, Yusong Tan
College of Computer
National University of Defense Technology
Changsha, China
{renyi, guanjb}@nudt.edu.cn

***Abstract*—Open source represents an important way in which today's software is developed. The adoption of open source software continues to accelerate because of the compelling economic and productivity benefits they provide. Therefore, it is important to mitigate potential legal exposure by scanning and analyzing the source of code. However, it is not the only benefit of this kind of work. While the complexity and the size of software composition grow, it is also critical to analyze the source of code to ensure code traceability and manageability, especially for software with tremendous scale of code, such as operating system. In this paper, we argue that it is beneficial for code audit and code quality insurance to distinguish the source of system packages and manage them separately. Then we propose an efficient method for code source classification based on package change log info extraction. And we design and implement KyAnalyzer, which support source analyzing and distinguished management of Debian/RPM packages with the proposed method. Preliminary experimental results show the correctness and efficiency of the package source classifier.**

Keywords—package source classifier, change log, open source, mixed source, operating system

I. 引言

开源软件是指源代码可以被自由获取和传播的软件，在随源代码开放的开源许可证协议规定范围内，允许被许可人对该软件进行研究、复用、修改、扩充和传播。开源软件已形成相对完整的软件体系，采用开源软件可有效提高软件开发效率和初始质量。目前，开源软件正在逐步改变全球软件生产和部署的格局，面向细分而多变的应用需求，借鉴和继承已有开源软件，快速搭建面向新需求的软件解决方案成为一种趋势[1-3]。例如，以 Amazon、百度、腾讯、阿里为代表的互联网公司分别基于开源 Linux 系统构建了其各具特色的应用解决方案，IBM SmartCloud Orchestrator 采用开放云架构，积极吸纳 Open Stack 等开源社区成果，作为 IBM SmartCloud 的技术基础。

尽管采用开源软件会带来一些潜在的安全、可靠、稳定以及软件版本依赖等问题，但可结合具体需求与发现的问题，通过分析、再设计、代码改造、扩充及调试以寻求有效的解决方案。国内外操作系统厂商也将开源软件纳入其产品研发的实践中。例如，苹果公司的 Mac OS X 和 iOS 操作系统产品是基于稳定的开源 BSD 内核设计开发的，而谷歌的 Android 系统则是借鉴了 Linux 内核，并面向智能终端对内核进行了深度裁剪定制和针对性增强，去除不必要的软件包，修正 Bug，并针对智能终端需求新研了 UI、Java 虚拟机、简单 C 库、SDK，形成独立可控的 Android 操作系统分支。

开发商的操作系统产品往往既借鉴了开源软件的相关代码，又包括大量结合软硬件需求的自主研发和改进的代码，不同来源的代码结构和质量参差不齐，形成了独特的混源操作系统形态。一方面，开源代码有助于缩短开发周期、降低开发成本、提高软件竞争力，另一方面，混源也为操作系统组成软件的管理带来了新问题，其中之一表现为混源软件来源链具有复杂性和多样性，目前，针对大规模混源软件的不同来源组成，缺乏基于来源特征分析的自动分类管理，代码可追踪性有待提高。

II. 相关工作

目前，支持对软件包进行来源分析的代表性软件包括商用 Black Duck 软件[4]和由 HP 公司主导开发的开源软件 FOSSology[5]。

Supported by National Natural Science Foundation of China (60603063, 61502510).

其中，Black Duck 是适用于软件整个生命周期的一个完整的开源软件管理/自动化解决方案，Black Duck 收集了数千个网站的开源项目信息，通过与开源代码库中代码的特征比对，来发现被比较的代码是否来自开源软件，是否存在 License 合规性问题。

FOSSology 支持软件的 License 扫描、Copyright 分析、包头信息提取等功能。与 Black Duck 相比，FOSSology 是通过搜索和匹配源代码中的元信息，实现 License 等的分析，缺少软件特征的智能分析能力。这

上述两种软件支持开源软件的扫描和分析，不具备为操作系统开发者自动识别和判断操作系统中软件包是开源、混源、还是自研的这一能力。

尽管传统软件工程和开源软件的发展非常迅速，目前，针对大规模的混源操作系统，主要工作集中在缺陷预测、检测、恢复、处理等方向，在软件包一级进行开源、混源、自研软件包的区分识别和管理方面的研究仍比较欠缺。现有的工作还主要是采用人工统计和辅助分析的方法，不但效率较低，也容易出错。因此，需要一种方法能够对操作系统软件包的来源进行自动分析。

III. 软件包来源分类

目前操作系统已经形成了两大体系。一个是 Unix/Linux 体系，另一个是 Windows 体系。其中，Unix/Linux 体系的操作系统中，软件包是系统的重要组成，操作系统内核、核外软件以软件包的形式进行组织和管理。

对于基于开源软件发展而来的操作系统产品，根据系统中软件包的不同来源，可以将其软件包分为三类：开源软件包、混源软件包和自研软件包。这三类软件包各自的特点如下：

- 开源软件包： 其源代码对公众开放，在软件 license 允许的范围下，可以被公众使用、修改和分发。开源软件包代码质量参差不齐、存在潜在安全漏洞和隐患，存在软件技术支持缺失等问题。
- 混源软件包：基于开源软件包，根据特定需求，由研发团队内部开发人员对开源软件包进行了修改和二次设计与开发，软件包的代码中开源和自研代码并存。与开源软件包相比，内部开发人员对混源软件包的认知和理解程度相对更高。另一方面，此类软件包仍然存在使用开源软件的潜在问题。
- 自研软件包：是指根据系统需求，由研发团队内部开发人员自行设计、研发的软件包，对代码框架、组成、功能和工作机理有深入和细致的了解和掌握，具备二次开发和快速迭代与升级能力。

上述不同来源的软件包特性不同，对其进行分类组织和管理有利于明晰操作系统组成软件包的代码自主程度，有助于增强软件包演进的可追踪性。

IV. KyAnalyzer 的设计与实现

A. 基本思想

软件包是用来完成特定任务的一组程序。当前各类应用都以软件包的方式提供给用户。Linux 平台下，常用的软件包种类主要有 RMP 和 Debian 两种。每个软件包里都包含有软件的修改历史，即修改日志（changlog）。修改日志信息对于区分软件包来源有重要作用，也可以辅助软件修改者了解软件开发的历史和内容，辅助开发人员在后续软件使用或修改时做出决策[6-8]。

一般地，修改日志中主要包含文档修改作者的姓名、邮箱，修改的时间，修改的版本号以及修改的内容等信息。通过 RMP 或 Debian 命令获取修改日志信息，并自动转换格式生成包含开发者信息的修改日志。生成的修改日志由至少一个修改项组成，每一个修改项对应代码包的一个版本更新，且最新的版本更新对应的是修改日志中的第 1 项修改项，每当为修改日志增加新的修改项时，将新的修改项放在修改日志中的第 1 项，并将修改日志中原有的修改项的编号全部加 1，使得第 i 条修改项更新为第 i+1 条修改项。通过软件包的修改日志数据结构中的信息和预先设置的包含开发者的信息集合，采用比较修改日志中的修改者姓名、邮箱等信息的方法可有效判断软件包来源。

B. 模块组成

KyAnalyzer 为软件包开发者和维护者提供所需的基础环境，它由软件包源代码扫描分析、软件包分类标识与管

理两个模块组成，如图 1 所示。用户通过 Web GUI 与 KyAnalyzer 进行交互。

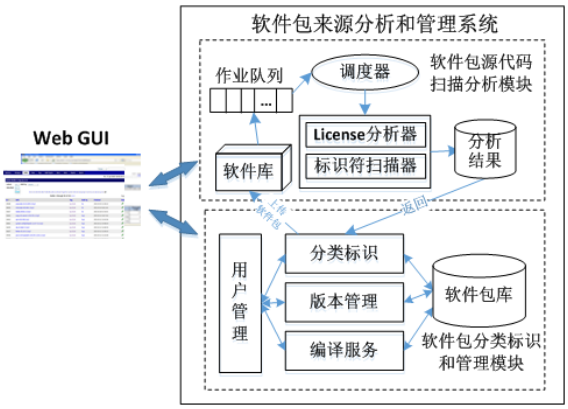


图 1 软件包来源分析和管理

软件包源代码扫描分析模块负责扫描提取软件包中的开源许可协议以及版权、开发者等标识符信息，该模块由软件库、作业队列、调度器、开源许可协议分析器、标识符扫描器、分析结果库六个部分组成，这六个部分的功能和交互过程如下：

- 首先，通过 Web GUI 将待扫描分析的软件包被上传到软件库中；
- 接着，根据要扫描的内容的不同，相应的任务加入到作业队列中；
- 调度器根据调度策略，从作业队列中选取要执行的任务，并根据任务的具体种类分发给开源许可协议分析器或者标识符扫描器；
- 开源许可协议分析器接收到调度器的调度请求后，负责分析提取软件包源代码中的开源许可协议信息，并将软件包中各个文件中可能发现的多种开源许可协议信息进行归纳、统计，将结果输出到分析结果库中；
- 标识符扫描器则负责从软件包的源代码中分析提取开发者电子邮箱、URL、版权等信息，并将结果输出到分析结果库中；
- 分析结果库负责存储开源许可协议分析器及标识符扫描器的输出结果。

软件包分类标识和管理模块负责对软件包进行分类、存储和管理，该模块由用户管理、分类标识、版本管理和软件包库组成。其中：

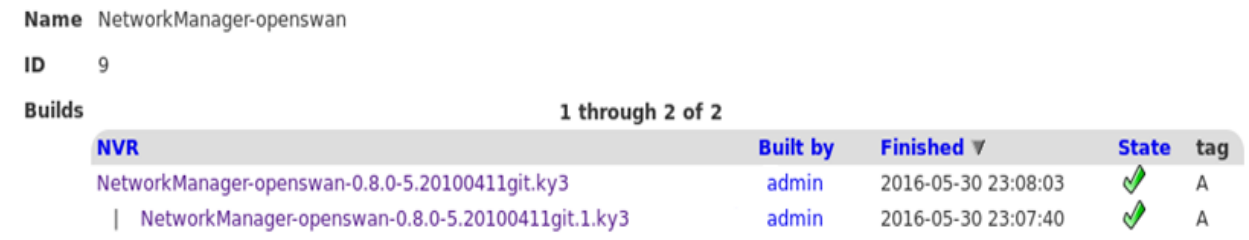
- 用户管理子模块支持多用户对软件包分类标识和管理模块的操作，提供多用户权限管理功能，用户通过该子模块登录并使用软件包分类标识和管理模块提供的分类标识、版本管理等各种功能，用户还可通过图形化界面浏览、查询、管理软件包。
- 分类标识子模块包括两个功能：一个是获取用户请求，调用软件包源代码扫描分析模块，将待分类软件包上传到软件包源代码扫描分析模块的软件库中，后者通过作业队列、调度器和标识符扫描器从该软件包中提取开源许可协议、开发者电子邮箱、URL、版权等数据并存入分析结果库中，并将结果返回给分类标识子模块，分类标识子模块根据返回结果，提取并按格式规整该版本软件包的修改日志，如果用户要求软件包入库，则将该包存入软件包库中；另一个是通过设计并实现的基于修改日志的操作系统软件包来源自动分类算法自动分析软件包代码来源并将其标记为开源、混源或自研软件包。
- 版本管理子模块负责软件包的版本管理，支持同一软件包不同版本之间演进关系的管理和图形化显示。
- 软件包库负责存储软件包，支持软件包名称、版本、来源分类等信息标识以及软件包的索引和查询，每个软件包都有一个唯一的 PackageID。

V. 实验和验证

软件包来源分析的流程如下：首先，上传需分析的软件源代码包；获取源代码包中的日志信息并修改生成包含开发者信息的修改日志；然后，基于修改日志中记录的开发者信息、修改项次数及预设的研发团队内部开发者信息或者开源项目开发者信息分析目标软件源代码包的软件包来源，得到为开源软件包、混源软件包、自研软件包三者之一的软件包来源分析结果；最后，为目标软件源代码包标记软件包来源分析结果。

本次实验以 RPM 包为例，验证运行在配置为 Intel(R) Core(TM) 2 Duo CPU T6570 @2.10GHz 2.10GHz，6.0GB 内存的计算机上，操作系统为 kylin 3.2。

测试对象包括系统常用软件包 995 个。任意选取系统中软件包上传到 KyAnalyzer 中进行分析。以 NetworkManager-openswan、NetworkManager 和 gltext 软件包为例，分析和标识结果如下：



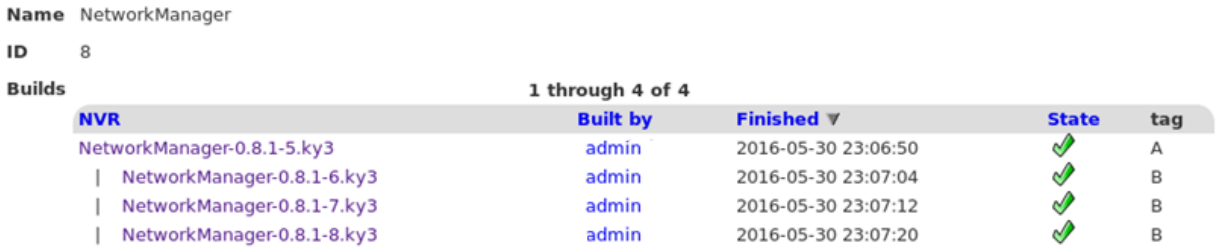
Name NetworkManager-openswan

ID 9

Builds 1 through 2 of 2

NVR	Built by	Finished	State	tag
NetworkManager-openswan-0.8.0-5.20100411git.ky3	admin	2016-05-30 23:08:03	✓	A
| NetworkManager-openswan-0.8.0-5.20100411git.1.ky3	admin	2016-05-30 23:07:40	✓	A

(a) 标记为 A-开源软件包



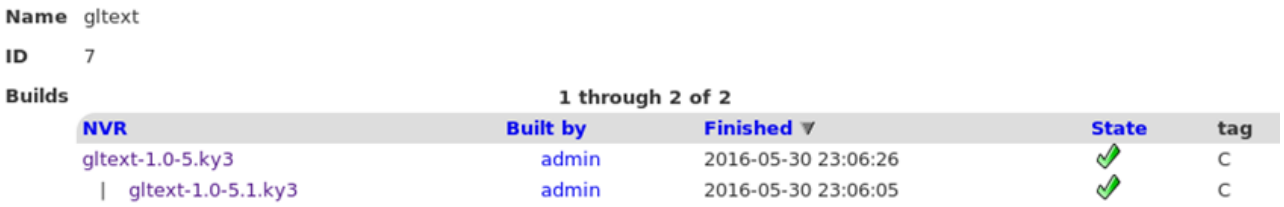
Name NetworkManager

ID 8

Builds 1 through 4 of 4

NVR	Built by	Finished	State	tag
NetworkManager-0.8.1-5.ky3	admin	2016-05-30 23:06:50	✓	A
| NetworkManager-0.8.1-6.ky3	admin	2016-05-30 23:07:04	✓	B
| NetworkManager-0.8.1-7.ky3	admin	2016-05-30 23:07:12	✓	B
| NetworkManager-0.8.1-8.ky3	admin	2016-05-30 23:07:20	✓	B

(b) 标记为 B-混源软件包



Name gltext

ID 7

Builds 1 through 2 of 2

NVR	Built by	Finished	State	tag
gltext-1.0-5.ky3	admin	2016-05-30 23:06:26	✓	C
| gltext-1.0-5.1.ky3	admin	2016-05-30 23:06:05	✓	C

(c) 标记为 C-自研软件包

测试输出的结果表明 KyAnalyzer 的软件包来源分类和标识功能正确、运行效率符合预期。

参考文献

[1] 倪光南. 开源软件在我国推广的机遇，挑战及其使命 [J]. 中国信息导报. 2006 (9):23-24

[2] 万江平，李德杰. 自由开源软件发展蓝图综述[J]. 计算机应用研究. 2009(11):43-45

[3] 浦靖珊. 开源软件的发展历史以及趋势分析[J].硅谷. 2014(9):4-4

[4] Black Duck Inc. Black Duck Hub: Find & fix open source vulnerabilities [EB/OL]. https://info.blackducksoftware.com/rs/872-OLS-526/images/BlackDuck_HUB_UL.pdf

[5] Get Started With FOSSology [EB/OL]. https://www.fossology.org/get-started

[6] Snipes W, Robinson, B.P.Murphy-Hill, E.R. Code hot spot: A tool for extraction and analysis of code change history[J]. ICSM .2011(9).14-17

[7] Adams B. Jiang, Z.M. Hassan A.E. Identifying crosscutting concerns using historical code changes[J]. ICSE. 2010(4):30-32

[8] Kagdi H, Collard M L, Maletic J I. A survey and taxonomy of approaches for mining software repositories in the context of software evolution [J]. Journal of Software Maintenance and Evolution: Research and Practice. 2007. 19(2): 77-131