Автоматизация управления компонентами с открытым исходным кодом при разработке программного обеспечения

Применение компонентов с открытым исходным кодом (open source) снижает стоимость разработки и ускоряет выпуск программ. Однако компоненты open source являются также и источником рисков для программного проекта. Во-первых, уязвимости безопасности могут использовать киберпреступники. Во-вторых, лицензионные соглашения open source бывают разных видов, и условия некоторых из них могут быть неприемлемыми для компании-разработчика. Таким образом, компоненты с открытым исходным кодом необходимо тщательно выбирать. Обо всем этом говорится в статье.

Д. А. ОРЕШКИНА, директор по развитию бизнеса Web Control

Многие компании разрабатывают программное обеспечение (ПО), одни для себя, другие для продажи, при этом почти все используют компоненты с open source – открытым исходным кодом. По данным известного аналитического агентства Gartner, еще два года назад 85% производителей программного обеспечения использовали open source, а более чем в 70% программного обеспечения имелись компоненты с open source.

Есть потребность в автоматизации процесса

Зачем писать код с нуля, если его уже кто-то написал, опубликовал и разрешил использовать в проектах? Однако то, что код уже протестирован и опробован, не значит, что он не содержит критичных ошибок и уязвимостей. Кроме того, множество компонентов с открытым исходным кодом имеют довольно строгие лицензионные требования – например, от вас могут потребовать бесплатно предоставлять исходный код своей разработки конечным пользователям, даже если ваш программный продукт распространяется за деньги.

Раньше, когда открытый код использовался довольно редко, разработчики отслеживали применяемые компоненты исходного кода вручную. Однако с ростом популярности open source в проектах обеспечивать таким способом соответствие стандартам и обнаруживать уязвимости компонентов становилось все более трудоемким. Теперь практически у всех разработчиков программного обеспечения имеется потребность в автоматизации процесса управления компонентами с открытым исходным кодом.

В 2002 г. стартап под названием Black Duck Software представил решение Protex, предназначенное для идентификации открытого исходного кода в программных продуктах. В основе его работы был сканер кода, который выявлял фрагменты исходного кода (code snippets), совпадающие с open source. Такой метод анализа компонентов имел большой процент ложных срабатываний (фрагменты проприетарного и коммерческого кода иденти-

фицировались как фрагменты открытого кода). В результате требовалась корректировка результатов в ручном режиме. Вскоре свои решения для сканирования кода стали предлагать и другие компании, такие как Protecode (сейчас являются частью Synopsys), Palamida (приобретена компанией Flexera Software) и Open Logic (приобретена компанией Rogue Wave).

Сканеры кодов были весьма дорогостоящими и трудоемкими, поэтому эту технологию для снижения рисков использовали в основном крупные компании.

Непрерывное управление открытым кодом

Использование метода непрерывного развертывания, увеличение доли open source в программах и повышение осведомленности об уязвимостях открытого кода изменили требования разработчиков. Написание программ посредством копирования фрагментов открытого кода со временем перестало быть распространенным, и поиск фрагментов кода стал менее востребованным.

Нужен был такой инструмент, который позволял бы обеспечивать безопасность кода и управлять лицензионными рисками в условиях agile-методологии разработки. И стала расти популярность систем класса Software Composition Analysis (SCA) для анализа компонентов ПО, которые доступны каждому.

Ценность систем SCA в том, что они позволяют обнаруживать уязвимости компонентов и/или управлять лицензионными рисками. При интеграции с инструментами разработки могут применяться политики SCA, чтобы автоматически блокировать проблемные компоненты еще до того, как они будут добавлены в проект. По данным компании The Forrester Wave, наиболее мощные решения в этом сегменте представлены производителями Black Duck Software (поглощена компанией Synopsys) WhiteSource Software, Sonatype, Flexera и Veracode. Некоторые из этих вендоров сделали акцент на управлении лицензионными рисками, другие – на вопросах безопасности. Система непрерывного управления открытым ПО имеет возможности автоматизации на протяжении всего цикла разработки программного обеспечения начиная со стадии выбора компонентов и заканчивая отслеживанием компонентов после выпуска релиза.

При выборе подходящего инструмента для управления open source обычно фокусируют внимание на следующих вопросах:

- инвентаризация компонентов;
- обнаружение уязвимостей;
- управление лицензионными рисками;
- возможности Shift Left и автоматизация.

Рассмотрим их более подробно.

Инвентаризация компонентов

Первый шаг в организации контроля лицензионных соглашений и уязвимости открытого кода – это инвентаризация бинарных компонентов и исходного кода open source в программном продукте. Для качественной инвентаризации важно, чтобы в решении SCA базы известного open source активно обновлялись.

Возникает вопрос: где оптимальнее всего обрабатывать полученные отчеты – в облаке или в собственном центре обработки данных (ЦОД)? Для одних компаний предпочтительнее SaaS-решение по подписке – это безопасно, удобно, можно быстро начать использование продукта. Для других компаний первостепенное значение имеет условие, чтобы никакая информация о проектах и используемых компонентах не покидала корпоративные границы – в таком случае требуется локальное (on-premise) решение, которое развертывается в корпоративном ЦОД или корпоративном облаке.

После того как инвентаризация всех применяемых компонентов open source проведена, необходимо получить информацию о рисках их применения.

Обнаружение уязвимостей

Постоянно сообщается о новых уязвимостях компонентов открытого кода. Киберпреступники знают, что уязвимость в одном компоненте может помочь им скомпрометировать большое количество систем. Необходимо проводить тщательную проверку безопасности загруженных библиотек и исходных кодов. Чем больше источников информации об уязвимостях используется и чем профессиональнее команда исследователей вендора проверяет эту информацию, тем мень-

Abstract. Usage of open source components decreases a cost of development and accelerates a software release. However, open source components are a source of risks for a software project as well. Firstly, security vulnerabilities may be intensively used by cyber criminals. Secondly, open source license agreements may be of different types, and the terms of some of them may be unacceptable for a software company. Thus open source components have to be carefully selected. This process is rather labour-intensive and requires.

Keywords. Analysis, open source, components software, vulnerability, quality, development.

Ключевые слова. Анализ, разработка, качество, уязвимости программного обеспечения, компоненты с открытым исходным кодом.

ше будет ложных срабатываний и тем качественнее результат проверки вашего ПО.

Выявить уязвимости – это только половина задачи, нужно еще принять ответные меры. Для ускорения процесса разработки развитые системы SCA позволяют автоматизировать формирование безопасного локального репозитория и выдачу рекомендаций разработчикам по устранению обнаруженных уязвимостей. Так можно существенно сэкономить время разработчиков и специалистов по безопасности.

Уже после выпуска релиза в ПО могут обнаружиться новые уязвимости. Система должна уметь отслеживать компоненты не только на этапе создания продукта, но и на всем его жизненном цикле. После выпуска релиза важно как можно скорее узнавать об уязвимостях в разработке.

Управление лицензионными рисками

Оценка риска нарушения лицензионных соглашений больше не является только частью юридических процедур в ходе слияния и поглощения крупных компаний или при выходе их на IPO. Постепенно она становится обычной процедурой перед каждый релизом ПО. Следовательно, юристам требуется автоматизированное решение для управления рисками лицензионных соглашений открытого кода, чтобы идти в ногу с непрерывным внедрением программных продуктов.

Отслеживание лицензий открытого кода в разрабатываемом продукте усложняется из-за расширения списка типов лицензий открытого ПО, множественности версий и огромного объема кода, опубликованного без упоминания лицензий. Системы SCA помогут при управлении лицензионными рисками путем автоматизированного создания отчетов правовой оценки с перечнем лицензий для юридического отдела и посредством применения лицензионной политики во время разработки ПО.

Возможности Shift left и автоматизация

Чем раньше обнаруживается проблемный компонент, тем легче и дешевле заменить его. Так называемое раннее обнаружение (англ. Shift left) заключается в проведении оценки ПО на наиболее ранних стадиях процесса разработки, что позволяет выявлять проблемы тогда, когда их легче и дешевле устранить. Отсев open source с критичной уязвимостью или неподходящими условиями лицензионного соглашения перед добавлением в свой репозиторий, сборку или даже перед началом тестирования ПО позволит повысить производительность ценных специалистов и улучшить качество реализованного проекта.

В системах управления open source для реализации раннего обнаружения и автоматизации предусмотрены инструменты анализа компонентов перед загрузкой, предупреждения о проблемах качества и безопасности open source в режиме реального времени, имеются возможности интеграции со средой разработки – репозиториями, системами сборки и CI-серверами[1]. Развитые SCA не просто оповещают о проблемах, но и имеют инструменты выбора наиболее подходящих компонентов.

Автоматизация – краеугольный камень CI/CD[2]. При этом автоматизированная система управления открытым ПО является критически важным элементом. Вручную невозможно обнаруживать проблемы с безопасностью, совместимостью и качеством в режиме реального времени. Автоматизация этих процессов не только защищает от человеческих ошибок и повышает точность, но и ускоряет процесс разработки и экономит драгоценные ресурсы.

Ключевые возможности систем управления open source в области автоматизации – блокирование использования проблемных компонентов, автоматическое применение политик, предупреждение об уязвимых и устаревших компонентах, развитый API (сокр. с англ. Application Programming Interface – интерфейс прикладного программирования) для интеграции со средой разработки, включая баг-трекеры. Наглядные отчеты о рисках open source в проектах – важный инструмент анализа эффективности SDLC[3] для руководителей.

ВЫВОДЫ

Ценность систем управления open source – в точном определении компонентов с открытым исходным кодом и их уязвимостей, автоматизации создания безопасного репозитория, автоматизации проверки компонентов open source на всем цикле разработки и сопровождения ПО и их лицензионных соглашений. Все эти возможности помогают сэкономить время ценных специалистов – разработчиков, тестировщиков, юристов, сотрудников отдела безопасности и управления рисками.

[1] Сервер непрерывной интеграции (сокр. с англ. Continuous integration server) – выделенный сервер, на котором организуется служба, в задачи которой входят: получение исходного кода из репозитория, сборка проекта, выполнение тестов, развертывание готового проекта, отправка отчетности.

[2] Непрерывная интеграция и развертывание программного обеспечения (сокр. с англ. Continuous integration/continuous delivery).

[3] SDLC (Software development lifecycle) – это серия из шести основных фаз, через которые проходит любая программная система. http://devprom.ru/news/SDLC-жизненный-цикл-разработки-системы.