6 мифов о статическом анализе кода

1 Миф

Внедрение статического анализатора в уже выстроенный цикл автоматической сборки кода и публикации полученного приложения может сильно затормозить эти процессы.

На самом деле

При использовании зрелых решений по анализу кода с возможностью тонкой настройки, которая проводится по утвержденной в организации методике, процессы сборки и публикации приложения не будут нарушены.

2 Миф

Чтобы анализировать код, нужно быть опытным программистом или этичным хакером.

На самом деле

Если вы используете базовые инструментальные средства по анализу кода, то работа с ними действительно может потребовать от вас специфических навыков и знаний.

Если же вы пользуетесь зрелыми программными продуктами с понятным интерфейсом, мощным аналитическим аппаратом и подробной сопроводительной информацией по каждой из выявленных уязвимостей, то специальные знания или опыт разработки вам не понадобятся.

3 Миф

Нет смысла проводить анализ кода на уязвимости, если его разработчики недоступны, – в этом случае не получится самостоятельно применить результаты проверки кода и исправить выявленные уязвимости.

На самом деле

При анализе вашего ПО, разработанного подрядчиком, с помощью зрелого анализатора кода формируется отчет со всей необходимой информацией и инструкциями по настройке WAF и другими мерами для минимизации уязвимостей. Все эти меры можно внедрить самостоятельно, без привлечения разработчиков.
Также есть возможность передать информацию об этих уязвимостях во ФСТЭК России, чтобы они связались с разработчиками – для этого есть специальная форма на ресурсе БДУ ФСТЭК. Эта опция возможна при подтверждении уязвимостей на стенде.

4 Миф

Невозможно провести анализ безопасности приложения, если нет доступа к его исходному коду.

На самом деле

Некоторые анализаторы кода дают возможность статического анализа бинарных файлов даже без доступа к исходному коду, вплоть до анализа мобильных приложений по ссылке из магазина приложений (Google Play и App Store).

5 Миф

После анализа будет получен огромный список уязвимостей и недостатков кода с большим количеством ложных срабатываний, на самом деле не требующих внимания.

На самом деле

В качественных решениях по анализу кода применяются эффективные алгоритмы и правила, дающие более точный результат, чем бесплатные или более простые анализаторы. Например, в статическом анализаторе кода Solar appScreener есть собственная разработка Fuzzy Logic Engine, которая минимизирует количество ложных срабатываний благодаря более гибкой настройке при проведении анализа.

6 Миф

Регуляторы не требуют проверки кода, поэтому для нашей организации эта задача не актуальна.

На самом деле

- Вопрос анализа кода важен для многих компаний, если они в своей деятельности подотчетны Банку России (согласно 683-П, 719-П и 757-П).
- Также с 1 января 2023 года он станет важным для организаций, разрабатывающих ПО для ОКИИ (согласно актуальным правкам к Приказу ФСТЭК № 239).
  Для многих вопрос применения статических анализаторов становится актуальным после получения письма из ФСТЭК России с требованием выполнять проверки Open Source с применением статического анализа.

Таким образом, анализ кода оказывается одной из важнейших мер, которые государственные регуляторы рекомендуют все чаще, и даже если ваша организация пока еще не попадает под требования регуляторов, то вполне возможно, что вскоре они вас коснутся. В таком случае стоит сыграть на опережение и реализовать эти меры у себя, основываясь на лучших мировых практиках.

Протестируйте решение Solar appScreener бесплатно!

Оставьте заявку на сайте и получите бесплатный демодоступ.

Оставить заявку

Ростелеком
Солар