УДК 004.056

ПРИМЕНЕНИЕ ОБЩЕЙ СИСТЕМЫ ОЦЕНКИ УЯЗВИМОСТЕЙ CWSS ПРИ РАЗРАБОТКЕ МЕТОДА ИНТЕГРАЛЬНОЙ ОЦЕНКИ УЯЗВИМОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ СТАТИЧЕСКОГО АНАЛИЗА

В.В. Семенов
(Санкт-Петербург, Университет ИТМО)
Научный руководитель – к.т.н. Гирик А.В.
(Санкт-Петербург, Университет ИТМО)

Существенное количество уязвимостей в информационных системах обусловлено недостатками в исходном коде программного обеспечения, входящего в их состав. Для крупных промышленных систем, разрабатываемых в основном на языках программирования высокого уровня (например, Java), ручной обзор кода и оценка уязвимости становится нетривиальной задачей. В связи с этим возникает потребность автоматизированного или автоматического поиска уязвимостей уровня исходного кода и общей формальной оценки качества кода. Один из ведущих способов автоматизации такого рода задач – статический анализ исходного кода, т.е. автоматический анализ кода без его исполнения. Современные инструменты статического анализа кода на Java, такие как Findbugs, SpotBugs, PVS-Studio способны находить многие недостатки и уязвимости уровня исходного кода, однако они не дают формальной численной оценки обнаруженных уязвимостей, необходимой для расчета общей интегральной оценки уязвимости ПО. По результатам обзора источников о формальной оценке уязвимостей, было обнаружено, что специалисты Института SANS и компании MITRE в сотрудничестве с экспертами по информационной безопасности США и Европы составили рейтинг наиболее опасных уязвимостей, ранжированных согласно формальной оценке по системе Common Weakness Scoring System (CWSS). Результат их работы – оценка отдельных уязвимостей. В рамках данного исследования показано, что система CWSS может быть расширена для применения в качестве основы для разработки метода совокупной (интегральной) оценки уязвимостей крупного программного проекта при осуществлении статического анализа.

Целью работы является разработка метода интегральной оценки уязвимостей на основе статического анализа исходного кода программного обеспечения на Java и реализация прототипа программы, реализующего разработанный метод.

Базовые положения исследования. В качестве ядра анализатора был выбран инструмент SpotBugs с установленным плагином find-sec-bugs. Были найдены соответствия кодов CWE и кодов правил обнаружения уязвимостей выбранного инструмента. Для этого проанализирован исходный код плагина и анализатора. В результате были обнаружены соответствия ряда кодов find-sec-bugs с идентификаторами CWE. В исследовании были определены наиболее опасные уязвимости программного обеспечения на языке программирования Java и рассчитаны их весовые коэффициенты. Каждый вид доступной для обнаружения уязвимости CWE получил оценку, соответствующую рейтингу SANS/MITRE. Для уязвимостей, отсутствующих в рейтинге, рассчитаны значения по системе CWSS. Система CWSS подразумевает четыре метода оценки: целевой, обобщенный, адаптированный к контексту и агрегированный. В рамках данной работы значения были рассчитаны как обобщенные, т.е. по видам уязвимостей. Это позволит использовать оценку с любым проектом на языке Java.

Результаты были приведены в формализованный вид и сведены в файл формата XML, затем переданы разработанному приложению. С учетом полученного перечня уязвимостей с

весовыми коэффициентами, была составлена формула расчета численного показателя интегральной оценки уязвимостей. Полученный показатель предназначен для сравнительного анализа общего уровня уязвимости программного проекта. Разработанный метод расчета был реализован в виде прототипа Java-приложения с графическим интерфейсом.

К разрабатываемому приложению был выдвинут ряд общих требований, включающий агрегацию данных по отчетам анализатора, загрузку отчетов из нескольких файлов заданной директории, наличие графического пользовательского интерфейса и отображение результатов работы в табличном виде.

Результаты работы. Разработан метод интегральной оценки уязвимости исходного кода программного обеспечения на языке программирования Java. Предложенный метод реализован в виде прототипа приложения для оценки уязвимости исходного кода. Приложение испытано на программном проекте с кодовой базой порядка 500 тысяч строк кода.