СТАТИЧЕСКИЙ АНАЛИЗ ИСХОДНЫХ КОДОВ В ОЦЕНКЕ НАДЕЖНОСТИ WEB-ПРИЛОЖЕНИЙ

Д.Е. Оношко, В.В. Бахтизин

Белорусский государственный университет информатики и радиоэлектроники, Минск, Республика Беларусь, onoshko@bsuir.by

Abstract. An overview of the most important trends in the field of software reliability and security is presented. An automatic vulnerability detection approach is proposed. A means of expressing the reliability and security characteristics of a web-application in the numeric form for further analysis is provided.

Являясь одной из форм образовательного процесса, наиболее активно развиваемых в настоящее время, дистанционное обучение предоставляет новые способы передачи знаний и позволяет детальнее рассматривать некоторые аспекты изучаемых предметов, ранее остававшиеся за рамками учебных программ. Одним из таких аспектов является актуальная в настоящее время проблема оценки, контроля и повышения качества web-приложений, рассмотрение которой при классической форме образовательного процесса традиционно носило обзорный характер.

В настоящее время программные средства (ПС) всё чаще разрабатываются как web-приложения, поскольку это позволяет получать доступ к таким ПС из любой точки мира посредством сети Internet. Это преимущество web-приложений также широко используется в реализации разнообразных образовательных проектов, направленных на организацию дистанционного обучения, в том числе и на международном уровне. Вместе с тем, всеобщая доступность таких приложений предопределяет повышенные требования к их качеству и особенно надёжности.

По данным Open Web Application Security Project (OWASP), по состоянию на 2013 год наиболее критичной проблемой безопасности web-приложений является их уязвимость к целому классу атак, которые принято называть «инъекциями» [1]. Их суть заключается в том, что уязвимое приложение позволяет злоумышленнику внедрить некоторый код, выполнение которого не предусмотрено разработчиком web-приложения. Эксплуатация таких уязвимостей в зависимости от их характера, области использования web-приложения и ряда других факторов может приводить к широкому спектру последствий от однократного отказа до утечек конфиденциальной информации, модификации финансовых и других важных данных, выхода из строя программных комплексов и т.д.

Среди приведённых в отчёте OWASP разновидностей атак-инъекций наибольшее количество web-приложений охватывает атака SQL-injection. Анализ web-приложений, уязвимых к данному виду атак, показывает, что причиной уязвимости является недостаточная и/или некорректная фильтрация данных, поступающих в web-приложение извне (от пользователя). Поскольку требования к обработке данных могут существенно отличаться даже в пределах одного web-приложения, разработка универсальных механизмов фильтрации и их использование на уровне языка программирования не представляются возможными. Единственным способом устранения уязвимостей данного класса остаётся полный анализ исходных кодов.

Поскольку полный анализ исходных кодов сравним по трудоёмкости с процессом разработки web-приложения, предпринимаются попытки разработки ПС, обеспечивающих автоматизированное обнаружение уязвимостей. Реализуемые в них методы анализа основаны на поиске сигнатур (потенциально опасных фрагментов) в формируемых SQL-запросах или исходном коде самих web-приложений. Между тем, такой подход эффективен для выявления только некоторого подмножества атак

SQL-injection. При этом уязвимости к атакам, использующим неизвестные разработчикам ПС подходы, не выявляются.

Для повышения результативности анализа предлагается подход, основанный на назначении оценок основным элементам web-приложения: переменным и процедурам. В общем случае оценки являются бинарными: «опасный» или «безопасный». Переменным назначаются оценки по следующим правилам:

переменные, в которые помещаются принятые от пользователя данные, получают оценку «опасный»;

переменные, в которые помещаются данные, прошедшие надлежащую фильтрацию, получают оценку «безопасный».

Процедуры рассматриваются как преобразования, выполняемые над данными. Операторы языка в рамках предлагаемого подхода являются процедурами, их операнды — параметрами, а результат выполнения оператора — возвращаемым значением. Формальные параметры процедур оцениваются следующим образом:

оценка «опасный» назначается тогда и только тогда, когда передаваемые в качестве параметра данные, не прошедшие фильтрацию, не приводят к появлению уязвимости в приложении;

оценка «безопасный» назначается для всех остальных параметров.

Оценка корректности вызова процедуры при этом сводится к сравнению оценок для формального и фактического параметров. Результаты выполнения процедур (т.е. возвращаемые значения) оцениваются аналогично. Для стандартных процедур оценки параметров назначаются в соответствии со спецификацией языка. Оценка остальных процедур производится путём проверки корректности составляющих их операторов.

Анализ web-приложения при этом производится путём построения по его исходному коду ориентированного графа, вершинами которого являются имеющиеся в web-приложении процедуры, а рёбрами — вызовы этих процедур. Само web-приложение рассматривается как процедура и целью работы алгоритма является получение оценок для её параметров и возвращаемого значения.

Для формализации получаемых анализатором результатов предлагается ввести понятие «точка входа данных» и определить его как семантически неделимую единицу данных, поступающих в оцениваемое web-приложение извне. Примерами таких точек входа могут быть поля ввода имени пользователя, пароля, текстов сообщений и т.д. При этом анализатор должен не только обнаруживать уязвимости, но и определять множество точек входа, которые могут быть использованы злоумышленником для их эксплуатации. Тогда в качестве одной из числовых характеристик надёжности ПС может использоваться отношение количества неэксплуатируемых точек входа к их общему количеству.

Предлагаемый подход к оценке надёжности позволяет получать числовые характеристики, не зависящие от размеров оцениваемого ПС, и, следовательно, может применяться для оценки надёжности ПС на всех этапах жизненного цикла начиная с момента появления первого прототипа, а также при сравнении различных ПС по уровню надёжности.

Литература

1. OWASP Top 10-2013. The Ten Most Critical Web Application Security Risks [Электронный ресурс]. — Режим доступа: http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf. — Дата доступа: 31.10.2013.