автоматизированного тестирования.

Механизм планирования и оценки покрытия разрабатываемого ПС тестированием может быть выбран в соответствии с потребностями системы и имеющейся документации. С учетом этого выбор применяемых методов и техник тестирования строго не регламентируется. План, методы и техники тестирования могут определяться командой на этапе планирования, исходя из субъективных мнений, предпочтений, доступных ресурсов [3]. Таким образом, задача формирования полного неизбыточного плана работ по КК разработки, пригодного для внедрения в проекты, процессы которых выстроены с использованием популярных моделей и методологий разработки на базе некоторых систем планирования процессов, может быть рассмотрена с аналитической точки зрения.

Отдельно при разработке в каждый момент времени стоят следующие задачи:

- оценка количества ошибок, оставшихся в проекте, и их критичность;
- оценка количества ошибок, оставшихся в выбранном компоненте, и их критичность;
- оценка времени, за которое текущая версия станет стабильной, т.е. количество ошибок и их критичность в данной версии будут меньше заданного порога.

Для решения данных задач можно использовать два подхода – изучение исходного кода программного продукта и применение моделей оценки надежности (МОН) ПС. Использование МОН позволяет посредством построения вероятностной модели случайных процессов и использования различных статистических методов получить оценки различных метрик (ожидаемое число ошибок, вероятность ошибки).

В докладе рассматривается возможность использования техник вероятностного (статистического) тестирования с целью создания репрезентативного сценария тестирования. В качестве некоторого критерия оценки эффективности использования данных техник тестирования можно принять достижение надежностью некоторого приемлемого для конкретного проекта значения [4].

На основе полученного сценария и располагая некоторыми данными для анализа с места планируемого внедрения плана работ, предлагается более предметно рассчитывать рентабельность автоматизации работ в процессе тестирования с учётом планируемой к использованию модели/методологии разработки с точки зрения затрат финансов и времени [2].

Списокиспользованныхисточников:

1. Royce, W. W. Managing the Development of Large Software Systems / Winston W. Royce // Article / Proceedings of IEEE WESCON 26 : Article / The Institute of Electrical and Electronics Engineers, Inc. – 1970.
2. Extreme Programming Explained: Embrace Change, 2nd Edition: Book / Addison-Wesley. – NY, 2004.
3. Липаев, В. В. Человеческие факторы в программной инженерии: рекомендации и требования к профессиональной квалификации специалистов : Учебник / В. В. Липаев. – М.: Синтег, 2009. - 328 с.
4. Sayre, K. Improved Techniques for Software Testing Based on Markov Chain Usage Models: Dissertation / K. Sayre. – Knoxville: The University of Tennessee, 1999. –128c.

ОПТИМИЗАЦИЯ АВТОМАТИЗИРОВАННОГО ОБНАРУЖЕНИЯ УЯЗВИМОСТЕЙ К SQL-ИНЪЕКЦИЯМ В WEB-ПРИЛОЖЕНИЯХ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Оношко Д.Е.

Бахтизин В.В. – к.т.н., доцент

Широкое распространение web-приложений как способа реализации программных средств, в сочетании с их доступностью неограниченному кругу пользователей предъявляет повышенные требования к качеству web-приложений. По данным OWASP [1], по состоянию на 2017 г. среди уязвимостей web-приложений наиболее распространёнными остаются SQL-инъекции. Обнаружение таких уязвимостей вручную является трудоёмким процессом, что создаёт необходимость в разработке методов их автоматизированного обнаружения.

Предложенная в [2] модель обнаружения уязвимостей предполагает выделение в web-приложении процедур, параметры каждой из которых в дальнейшем подвергаются оценке. Однако, несмотря на относительную простоту модели, ввиду значительного объёма исходных кодов, присущего современным web-приложениям, при реализации метода обнаружения уязвимостей на основе этой модели возникает необходимость оптимизации процесса назначения оценок.

Очевидным способом оптимизации является исключение из рассмотрения тех частей web-приложения, которые не оказывают влияния на результаты обнаружения уязвимостей. В основу такой оптимизации предлагается положить разделение модели на 2 уровня: внутрипроцедурный и межпроцедурный.

Межпроцедурный уровень модели предлагается представить в виде графа зависимостей, отражающего характер взаимосвязей между отдельными процедурами web-приложения. Упрощённый пример такого графа представлен на рисунке 1.

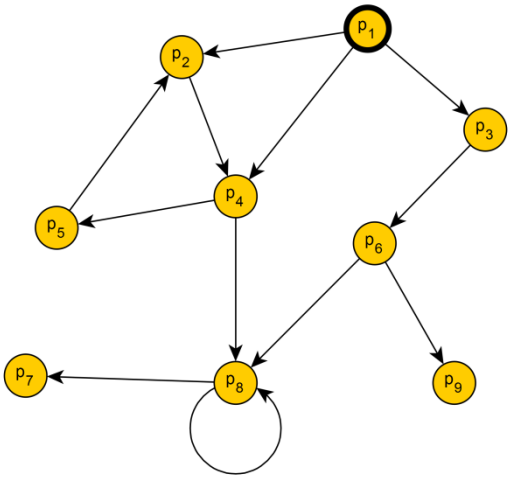

Рис. 1 – Упрощённый пример графа зависимостей

Такое представление структуры web-приложения позволяет очевидным образом:

- определять порядок анализа отдельных процедур web-приложения;
- выявлять взаимную рекурсию с целью исключить зацикливание программного средства, реализующего метод обнаружения уязвимостей, — анализатора;
- исключать из рассмотрения недостижимый код.

Начинать анализ следует с процедур, не имеющих исходящих дуг в графе зависимостей. В примере это процедуры p_7 и p_9. Точка входа web-приложения p_1 (на рисунке выделена) подвергается анализу в последнюю очередь, а полученные её параметрами оценки позволяют дать ответ на вопрос о фактическом наличии или отсутствии в web-приложении уязвимостей к SQL-инъекциям.

Выделение межпроцедурного уровня позволяет оптимизировать процесс обнаружения уязвимостей за счёт отказа от включения во внутрипроцедурный уровень модели web-приложения информации о ряде процедур, а также выбора оптимального порядка накопления такой информации. При наличии соответствующей поддержки со стороны анализатора также представляется возможным распараллеливание анализа процедур, не имеющих взаимных зависимостей.

Список использованных источников:

1. OWASPTop 10 2017. The Ten Most Critical Web Application Security Risks. [Электронный ресурс.] — Режим доступа: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf. — Дата доступа: 27.11.2017.
2. Бахтизин В. В. Модель обнаружения уязвимостей в web-приложениях / В. В. Бахтизин, Д. Е. Оношко // Докл. БГУИР. — 2016. — №1 (95). — С. 5–11.

ВИЗУАЛИЗАЦИЯ БОЛЬШИХ ДАННЫХ (BIGDATA)

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Панкратьев А.С., Пилинко Н.А., Голубев К.А.
Лапицкая Н.В. – к.т.н, доцент

В настоящее время объем необходимых для обработки данных стремительно растут, однако общеупотребительные методы их представления и обработки требуют много ресурсов и времени, а также недостаточно просты для понимания. Одним из решений этой проблемы может стать представление данных в графическом виде на пространственно-временной поверхности.

Цель: представить массивные структуры данных на пространственно-временной поверхности.

Рассмотрим использование данного метода на примере представлении комплексной статистики по пожарам на карте.

Входными данными будем считать:

- База данных МЧС РБ по пожарам за последние 11 лет;
- GET-параметры с выбранными пользователем фильтрами;

В пространственной плоскости S данные представляются согласно административно-территориальному делению.

Разобьем плоскость на m частей, где Si - представление на карте частотной и количественной информации за период.

Каждый Si является функцией. Si = f (x, y, z, c), где x, y, z - географические координаты, c - полученный