Л.В. Рудикова / Доклады БГУИР. - Мн. : БГУИР, 2014. - №1 (79). - С.46-52.

АНАЛИЗ ЗАЩИЩЕННОСТИ WEB-ПРИЛОЖЕНИЙ

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

Халецкий С.Д.

Глухова Л. А. – канд. техн. наук, доцент

В докладе обоснована значимость анализа защищенности web-приложений, а также рассмотрены методы повышения защищенности путем своевременного обнаружения и устранения недостаточно защищенных web-приложений.

Защищенность – степень защиты программным продуктом или системой информации и данных так, чтобы люди, другие продукты или системы имели степень доступа к данным, соответствующую типам и уровням их авторизации.[1]

К подхарактеристикам защищенности относят: конфиденциальность, целостность, неопровержимость, идентифицируемость, аутентичность.

Зачастую современные web-приложения имеют дело с конфиденциальной информацией, которая в свою очередь доступна посредством Web. При этом обмен информацией между браузером и сервером происходит по открытым каналам с использованием открытых протоколов. В связи с этим контролировать передаваемые данные сложно. Поэтому важное значение имеют вопросы обеспечения защищенности web-приложений.

Open Web Application Security Project (OWASP) – это библиотека, содержащая исчерпывающее руководство по поиску различного рода уязвимостей, а также содержит рекомендации к процессу проведения анализа защищенности web-приложений.

Наиболее простым и, как следствие, более распространенным способом анализа защищенности web-приложений является инструментальное обследование. Данный метод предусматривает использование сканеров безопасности, а также дополнительных инструментов, автоматизирующих некоторые сценарии эксплуатации и выявления уязвимостей. Одним из основных недостатков данного подхода является то, что необходимо постоянно поддерживать сигнатуры в актуальном состоянии, а также для получения корректной оценки работы необходимо осуществлять проверку транзакционно, то есть на ряду с проверкой результатов последнего вызова, проверять результаты предыдущих, что является весьма непросто.

Использование инструментального анализа не всегда является возможным, например в банковской сфере. Важность выявления уязвимостей приводит к тому, что их поиск необходимо осуществлять также и вручную, хотя это и требует больших временных затрат, а также не исключает проявление “человеческого фактора”.

Если исходный код используемых приложений, сервисов, библиотек является открытым, достаточно действенным методом по обнаружению уязвимостей будет анализ исходного кода. Если не требуется проверка воспроизведения найденных уязвимостей, то анализ можно будет проводить вообще не затрагивая работу самого web-приложения. Наибольшее распространение среди методов по анализу исходного кода получил метод статического анализа, основанный на использовании сигнатур, базисом которых являются регулярные выражения. Так как не все сигнатуры могут присутствовать, то некоторые из уязвимостей будут не выявлены. Поэтому совместно со статическим анализатором кода следует применять динамические анализаторы, которые на низком уровне разбирают синтаксис языка программирования web-приложения, после проверок которого выявляются грубые ошибки, допущенные разработчиками. Стоит учитывать, что, наряду со сканерами, анализаторам присущи те же недостатки.

Организацию процесса анализа защищенности следует начинать прежде всего с постановки цели самого анализа, определения области исследования, после чего сформировать перечень производимых проверок. В зависимости от цели анализа выбирается стратегия. Если необходимо выявить возможности проникновения, нарушение штатного режима, то приложение следует рассматривать как “черный ящик”, проведение работ будет осуществляться без предварительного получения какой-либо информации о нем. Если достаточных средств для проведения анализа нет, а цель – повышение уровня защищенности web-приложения, то его стоит рассматривать как “серый ящик” (например, когда злоумышленнику известна структура каталогов, исходный код некоторых файлов, функций) с использованием инструментального подхода к его анализу, а также использовать ручные проверки в наиболее критических местах.

Так как web-приложения ориентированы на массовое использование, то сбои в работе, вызванные действиями злоумышленника, оказывают сугубо негативные последствия. Этому также способствует возможность использования однотипных сценариев эксплуатации уязвимостей, так как зачастую используются и шаблонные решения, а обновление до актуальной безопасной версии не осуществляется. Поэтому анализ защищенности web-приложения должен быть частью общей стратегии обеспечения защищенности.

Список использованных источников:

1. ISO/IEC 25010:2011. Проектирование систем и разработка программного обеспечения. Требования к качеству систем и

программного обеспечения и их оценка (SQuaRE). Модели качества систем и программного обеспечения. – Введ. 2011-03-03. – Женева, 2011.

СИСТЕМА КРИТЕРИЕВ ОЦЕНКИ ЭФФЕКТИВНОСТИ ПРОГРАММНЫХ СРЕДСТВ ОРГАНИЗАЦИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

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

Царевич Д.Ю.

Бахтизин В.В. – к.т.н., доцент

В связи с интенсивным развитием сфер человеческой деятельности, в том числе научной, экономической, медицинской, растут потребности общества в вычислительных ресурсах, необходимых для обработки информации. Одним из возможных способов удовлетворения данных потребностей является использование распределенных вычислений (РВ).

Для оценки и сравнения параметров программных средств организации РВ в работе формулируется система критериев оценки эффективности ПС организации РВ, включающая следующие критерии [1]:

1. Производительность – ключевой критерий, характеризуется отношением показателей производительности ПС организации РВ: максимального из полученных практически к максимально возможному теоретически.
2. Масштабируемость – способность ПС организации РВ выполнять работу, пропорциональную числу вычислительных узлов (ВУ) без потери производительности при росте числа ВУ.
3. Безопасность – описывает эффективность механизмов защиты информации, вычислительных узлов и процессов ПС организации РВ от несанкционированного доступа.
4. Отказоустойчивость – характеризуется соотношением промежутка времени, в течение которого ПС организации РВ находилось в состоянии работы к общему промежутку времени, в течение которого производилось измерение.
5. Прозрачность – характеризует способность ПС скрывать свою распределенную природу, т.е. распределение процессов и ресурсов по множеству ВУ, и способность представляться для пользователей и разработчиков приложений в виде единой централизованной компьютерной системы.

В качестве критерия оценки *производительности* ПС организации выбрано отношение значений двух используемых вариантов измерения производительности – максимальной и пиковой [2].

Пиковая производительность (P_{peak}, TFLOPS) – теоретический предел производительности (выражаемый через операции с плавающий точкой) для оцениваемого ПС, находится с помощью формулы (1.1):

$$P_{peak} = \sum_{i=1}^{N}(F_i * P_i * I_{tick} / C), \quad (1.1)$$

где F_i – тактовая частота процессора i ВУ, МГц.

P_i – число процессоров в i ВУ.

I_{tick} – количество инструкций с плавающей запятой на такт (4 – для процессоров Core2; 8 – для процессоров Intel с AVX).

C – константа, $C = 10^6$.

N – количество ВУ, используемых ПС.

Максимальная производительность – максимальная производительность ПС, достигаемая при решении практических задач.

Критерий оценки производительности P ПС организации РВ определяется как соотношение значений максимальной и пиковой производительностей по формуле (1.2):

$$P = P_{max} / P_{peak} * 100\%, \quad (1.2)$$

где P_{peak} – пиковая производительность, TFLOPS.

P_{max} – максимальная производительность, TFLOPS.

Рассматривая *масштабируемость* как способность оцениваемого ПС выполнять работу, пропорциональную числу вычислительных узлов (ВУ), критерий масштабируемости определяется с помощью критерия оценки параллельной эффективности [2].