Таким образом, предлагается архитектура подсистемы хранения в системах принятия решений, позволяющая улучшить количественные характеристики системы путём проектирования с учётом особенностей предметной области.

Список использованных источников:

1. Zaharia, M. An Architecture for Fast and General Data Processing on Large Clusters / M. Zaharia // PHD Dissertation. – 3 February 2014 / Berkeley, USA. – 2014. P. 3 – 35
2. Pu, Q. FairRide: Near-Optimal, Fair Cache Sharing / Q. Pu, H. Li, M. Zaharia, A. Ghodsi, I. Stoica // NSDI. – 16 March 2016 / Santa Clara, USA. – 2016. P. 2 - 3

ПОСТРОЕНИЕ АНАЛИТИЧЕСКОГО СЕРВИСА ДЛЯ АНАЛИЗА СПЕКТОРОВ

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

Усиков А. В.

Рудикова Л.В. – доцент, канд.физ-мат.наук

На сегодняшний момент накоплено довольно много материалов, которые представлены в определенных таблицах и атласах спектральных элементов. Непосредственная автоматизация процесса обрабатывания спектрограмм, получаемых с помощью мобильного лазерного спектрометра, представляет собой программные инструменты, которые включают в себя достаточно узкий комплект возможностей. Недоступность инструмента, который дает возможность хранить и анализировать спектры в режиме реального времени с интеллектуальным поиском является актуальным направлением.

Установим необходимые требования к разрабатываемому сервису. В первую очередь в сервисе должны присутствовать возможности аналитической обработки данных, интеллектуального поиска данных (алгоритмы добычи данных), использования нейронных сетей для прогнозирования, корреляций, типовых образцов и исключений в больших объемах данных спектров. Необходимо отметить, что каждая из этих возможностей представляет собой самостоятельные части с точки зрения архитектуры.

Сервис должен равным образом предоставлять удаленный доступ к проанализированным и результирующим данным, поэтому необходимо предоставить публичную конечную точку пользователям для доступа в любое время к сервису и получения данных анализа.

Многоуровневая архитектура обеспечивает группировку связанной функциональности приложения в различных слоях, выстраиваемых вертикально, поверх друг друга. Слои слабо связаны, и между ними осуществляется явный обмен данными. Точное разбивание приложения на слои помогает поддерживать строгое распределение функциональности, что в свою очередь, обеспечивает гибкость, а также практичность и несложность сопровождения.

Функциональные области приложения разделяются на многослойные группы (уровни). Сервис состоит из шести взаимодействующих друг с другом слоев: уровень хранения данных, уровень надстроек, уровень доступа к данным, уровень бизнес-логики, уровень сервисов, уровень клиентов Исходя из предъявляемых требований, для анализа спектров, полученных в результате лазерной экспрессной экспертизы будет приемлемым разрабатывать с применением многоуровневой архитектуры.

Для построения сервиса представлена многоуровневая архитектура. Подобная архитектура обеспечивает группировку связанной функциональности приложения в различных слоях, выстраиваемых вертикально, поверх друг друга. Слои слабо связаны, и между ними осуществляется явный обмен данными. Точное разбивание приложения на слои помогает поддерживать строгое распределение функциональности, что в свою очередь, обеспечивает гибкость, а также практичность и несложность сопровождения.

Каждый слой агрегирует ответственности и абстракции уровня, расположенного непосредственно под ним. При строгом разделении на слои составляющие одного слоя могут взаимодействовать только с составляющими такого же слоя или составляющими слоя, расположенного прямо под данным слоем. Более свободное разделение на слои позволяет составляющим взаимодействовать с составляющими того же и всех нижележащих слоев.

Внедрение подобного рода возможностей для обработки, анализа и хранения спектров, полученных в результате лазерной экспрессной экспертизы, дадут возможность внушительно усовершенствовать скорость аналитической обработки данных с интеллектуальным поиском, использованием нейронных сетей для прогнозирования сходств, корреляций, типовых образцов и исключений в больших объемах данных спектров. Следует отметить, что благодаря версионному хранению спектров, данные снимков каждый раз можно реконструировать до конкретного состояния в зависимости от требований..

Список использованных источников:

1. Кудрявцев, Ю. А. OLAP технологии: обзор решаемых задач и исследований / Ю.А. Кудрявцев // Бизнес-информатика. - 2008. - № 1. - С. 66-70.
2. Рудикова, Л. В. Разработка программного визуализатора спектров для поддержки лазерной экспрессной экспертизы //

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