

локального хоста). Привилегии пользователей выставляются на 4-х уровнях: глобальный уровень, уровень баз данных, уровень таблиц и уровень столбцов.

Исходя из вышеописанного, анализ безопасности в рамках СУБД MySQL может осуществляться путём проверки доступности пользователей, а также выявления небезопасных привилегий, которыми они обладают. Нарушением безопасности можно считать ситуацию, когда пользователь root доступен вне сервера (localhost) или, по крайней мере, вне локальной сети. Доступ к обычным пользователям (не root), в большинстве случаев, требуется обеспечить только с одного хоста или локальной сети, однако возможность доступа к пользователям с нескольких хостов нельзя трактовать как нарушение безопасности, но не будет лишним сформировать список с такими пользователями для изучения администратором.

СУБД MySQL допускает отсутствие пароля у пользователей. В связи с этим, необходимо выявить всех пользователей у которых не установлен пароль. При анализе привилегий пользователей, в первую очередь следует проверить привилегии глобального уровня. В большинстве случаев, наличие глобальных привилегий требуется только у пользователя root. Наличие глобальных привилегий у других пользователей представляет потенциальную опасность для системы. Наиболее опасны такие глобальные привилегии как: DELETE, DROP, SHUTDOWN, GRANT, ALTER, PROCESS, SUPER, LOCK_TABLES, EXECUTE, ALTER_ROUTINE, CREATE_USER. Доступ к системным таблицам, таким как mysql.users например, следует предоставлять только для пользователя root.

В качестве дополнительной меры обеспечения безопасности можно просканировать порты с помощью утилиты типа nmap. MySQL использует по умолчанию порт 3306. Этот порт должен быть недоступен с неблагоннадёжных компьютеров. Также для проверки открыт порт или нет, можно попытаться установить соединение через Telnet. Если соединение будет установлено, это будет означать, что порт открыт, и его следует закрыть на брандмауэре или маршрутизаторе (если, конечно, нет действительно веских причин держать его открытым) [1].

Список использованных источников:

1. mysql.ru [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.mysql.ru/>.
2. Кузнецов, М. В. MySQL 5 / М. В. Кузнецов, И. В. Симдянов. – СПб: БХВ-Петербург, 2010. – 1024 с.

АРХИТЕКТУРА ПРОГРАММНОГО СРЕДСТВА ВЕДЕНИЯ СТАТИСТИКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

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

Сачок Н. П.

Курмаз Ю. П. – ассистент кафедры ПОИТ

В настоящее время важной задачей разработки программного средства является его итерационное улучшение от выпуска к выпуску. Одним из методов определения необходимых улучшений является сбор и анализ статистики использования приложения.

В настоящее время одними из самых распространенных подходов к разработке ПО являются так называемые итеративные подходы. Кроме классических приемов, направленных на работу с заказчиком и итерационной реализации его требований для получения готового продукта, очень часто появляется необходимость продолжить итеративную разработку программного средства уже после его выпуска для конечного пользователя. Встает задача изучить процесс взаимодействия конечного пользователя и программного средства, выявить ошибки, сделанные на предыдущих этапах, а так же определить стратегию улучшения программного средства и его дальнейшего развития с учетом реальных запросов пользователей.

Основным способом решения поставленной задачи является сбор и анализ статистики использования приложения. В случае мобильного приложения, нацеленного на массового пользователя, в первую очередь интерес представляют следующие метрики использования программного средства:

DAU (daily active users), MAU (monthly active users) — метрики, которые показывают соответственно количество ежедневно и ежемесячно активных пользователей мобильного приложения.

ARPPDAU (**average revenue per daily active user - средний доход на одного активного пользователя в день**) — метрика, позволяющая оценить эффективность монетизации для freemium приложений. Показывает, какую в среднем прибыль приносит один активный пользователь приложения.

LTV (loan to value) — отношение общей прибыли от приложения к общему числу установок. Так же позволяет оценить финансовую эффективность приложения, но уже в течение времени жизни приложения.

Rolling Retention — целая группа метрик, которая в отличие от предыдущих привязана не к времени жизни приложения, а ко времени жизни какой-то группы пользователей (от первого открытия до последнего использования).

В процессе анализа требований к программному средству ведения статистики мобильного приложения выяснилось, что все необходимые метрики строятся на основе практически одних и тех же

данных, но представляют их с разных точек зрения, для облегчения анализа данных маркетологами и бизнес-аналитиками. Исходя из этого была предложена распределенная архитектура программного средства, изображенная на рисунке 1.



Рисунок 1. Распределенная архитектура программного средства

Данная архитектура обладает следующими достоинствами:

Стабильность. При выходе из строя сервера обработки статистики сбор пользовательских данных не будет прекращен, данные не будут утеряны и их обработка может быть продолжена при восстановлении работы сервера. При выходе из строя сервера сбора статистики, анализ статистики может быть продолжен с использованием старых данных до восстановления нормальной работы, что позволяет не создавать задержек в процессе работы бизнес-аналитиков.

Модульность. Разделение программного средства на два независимых сервера сбора и обработки статистики позволяет производить работы на одном из серверов, не затрагивая работоспособность другого, а так же при необходимости отключить и полностью заменить один из них. Кроме того при введении новых метрик использования приложения их можно рассчитать на всем наборе данных за все время жизни ПС.

Накопление опыта. Сервер обработки и анализа статистики и базу данных обработанной статистики можно использовать независимо, например, при закрытии проекта и прекращении сбора статистики, или передавать их копии другим командам, для анализа опыта предыдущих проектов, исследования удачных и неудачных решений и использования накопленного опыта в новых проектах.

Список использованных источников:

1. Куликов, С.С. Метод увеличения производительности интернет-ориентированных клиент-серверных приложений / С.С. Куликов, О.Г. Смолякова // Доклады БГУИР. – 2010. – № 3 (49). □ С. 81-86.
2. Критерии оценки успешности социальных игр [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://bankomet.com.ua/2012/01/kriterii-otsenki-uspeshnosti-sotsialnyh-igr/>
3. Про геймдев: онлайн и мобильные игры - монетизация, аналитика, growth hacking [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.progamedev.ru/2013/05/marketing-finance-friendship.html>

МОДЕЛИРОВАНИЕ СИСТЕМЫ ОБРАБОТКИ ПЛАТЕЖНЫХ ПАКЕТОВ В УСЛОВИЯХ МНОГОПОТОЧНОЙ СРЕДЫ

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

Мельников А.Ю.

Бахтизин В.В. – канд. техн. наук, доцент

Проведен анализ возможности модернизации модели работы платежной системы, основной задачей которой является конвертация платежных пакетов из формата платежа одной страны в другой формат. Предложен вариант рационального использования многопоточной среды.

Некоторые современные системы не учитывают особенности работы в условиях многопроцессорной обработки данных. В этой связи модель работы системы банковских расчетов должна быть подвергнута