ВЫЯВЛЕНИЕ НАРУШЕНИЙ БЕЗОПАСНОСТИ ПРИ АДМИНИСТРИРОВАНИИ СУБД MySQL

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

Навицкий И.П.

Куликов С.С.–к.т.н., доцент

В настоящее время в базах данных хранятся большие объёмы информации. Базы данных могут содержать множество таблиц сложной структуры для хранения и представления различного рода данных. По мере роста и развития баз данных, всё чаще возникает необходимость системного управления и администрирования, постоянного сервисного обслуживания, контроля и диагностики. Приложение, содержащее различные сервисные функции для работы с базами данных, помогает поддерживать их в рабочем состоянии, анализировать параметры безопасности и производительности, устраняя тем самым потенциальные проблемы и сбои в работе баз данных в будущем. Поскольку в базах данных может храниться важная информация, то крайне необходимо обеспечить корректную работу баз данных и сохранность информации.

В настоящее время довольно часто встречаются ситуации, когда на одном сервере располагается несколько баз данных, которые управляются одной локальной клиент-серверной СУБД (системой управления базами данных). СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, в монопольном режиме. Все клиентские запросы на обработку данных обрабатываются клиент-серверной СУБД централизованно.

Подключаться к серверу для работы с базами данных могут различные ресурсы, причём никак не связанные между собой. В связи с этим, становятся актуальными вопросы обеспечения безопасности и распределения доступа к данным.

Используемая в MySQL система безопасности для всех подключений, запросов и иных операций, которые может пытаться выполнить пользователь, базируется на списках контроля доступа ACLs (AccessControlLists) [1].

При обсуждении вопросов безопасности акцентируется внимание на необходимости защиты всего серверного хоста (а не одного лишь сервера MySQL) от всех возможных типов атак: перехвата, внесения изменений, считывания и отказа в обслуживании.

Основной функцией системы привилегий MySQL является аутентификация пользователя, подключающегося с указанного хоста, и ассоциирование его с привилегиями базы данных, такими как SELECT, INSERT, UPDATE и DELETE. Контроль доступа осуществляется с помощью трёх полей контекста таблицы user (Host, User и Password) [2]. На рисунке 1 представлена схема порядка проверки привилегий и получения доступа к базе данных.

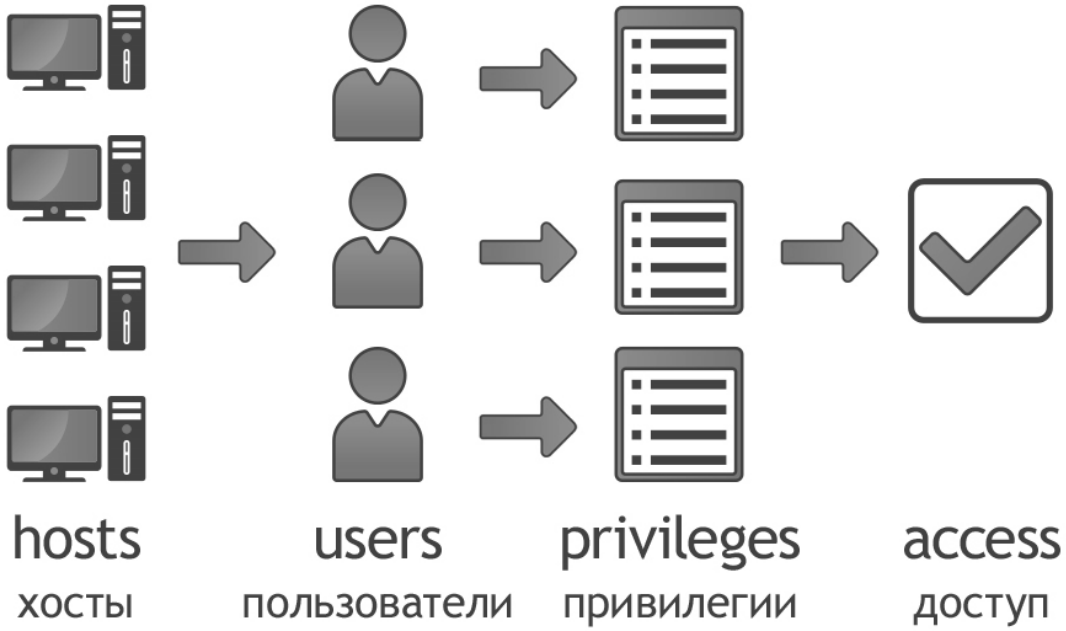

Рис. 1 - Порядок получения доступа

Для каждого пользователя в СУДБ MySQL задаётся его доступность с различных хостов. В качестве хоста может указываться его имя, IP-адрес, диапазон IP-адресов, либо значение 'localhost' (доступ с

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. Наиболееопаснытакиеглобальныепривилегиикак: 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 неблагонадёжных 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.

ARPDAU (**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