ИССЛЕДОВАТЕЛЬСКИЕ РАБОТЫ УЧАЩИХСЯ

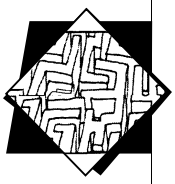
В разделе публикуются исследовательские работы школьников, выполненные в самых разных областях знаний. В журнале представлены исследования участников различных всероссийских конкурсов и конференций. Работы прокомментированы учёными-специалистами в данных областях науки. Цель комментария — обратить внимание читателя как на сильные, так и на слабые стороны публикуемой работы; на различные методические и содержательные аспекты проведённого исследования

Информационная безопасность баз данных. Платформа Oracle

Кирилл Бабеев,
ученик гимназии г. Обнинска
Научные руководители:
Александрова Татьяна Петровна,
преподаватель высшей категории;
Данюков Николай Иванович,
кандидат технических наук, старший консультант ОАО «Оракл»

Введение

Постоянные предложения приобрести различные базы данных свидетельствуют о том, что продажа конфиденциальных сведений о гражданах и юридических лицах стала отдельным видом бизнеса. Если появление очередной опубликованной базы для граждан является просто ещё одним малоприятным фактом обнародования сведений об их частной жизни, то в рамках некоторых предприятий это может отрицательно повлиять на бизнес. Например, для оператора сотовой связи распространение базы биллинга может обернуться существенным оттоком абонентов к более «надёжному» оператору-конкуренту. Поэтому оператору подчас экономически более выгодно найти «производителя», подготовившего украденную базу к продаже, и вы-



купить весь тираж. Но проблема перекрытия возможных утечек при этом остаётся весьма актуальной.

Защита баз данных является одной из самых сложных задач, стоящих перед подразделениями, отвечающими за обеспечение информационной безопасности. С одной стороны, для работы с базой необходимо предоставлять доступ к данным всем сотрудникам, кто по долгу службы обязан осуществлять сбор, обработку, хранение и передачу конфиденциальных данных. С другой стороны, укрупнение баз данных далеко не всегда имеет централизованную архитектуру (наблюдается ярко выраженная тенденция к территориально распределённой системе), в связи с чем действия нарушителей становятся всё более изощрёнными. При этом чёткой и ясной методики комплексного решения задачи защиты баз данных, которую можно было бы применять во всех случаях, не существует, в каждой конкретной ситуации приходится искать индивидуальный подход.

Классический взгляд на решение данной задачи включает обследование предприятия с целью выявления таких угроз, как хищение, утрата, уничтожение, модификация, отказ от подлинности. На втором этапе следует составление математических моделей основных информационных потоков и возможных нарушений, моделирование типовых действий злоумышленников. На третьем — выработка комплексных мер по пресечению и предупреждению возможных угроз с помощью правовых, организационно-административных и технических мер защиты. Однако разнообразие деятельности предприятий, структуры бизнеса, информационных сетей и потоков информации, прикладных систем и способов организации доступа к ним не позволяет создать универсальную методику решения.

Цели и задачи

Цели:

1. Показать возможные угрозы в СУБД.
2. Установить и продемонстрировать возможности Oracle Database Firewall.
3. Разработать безопасную архитектуру для работы с базами данных.
4. Проверить работоспособность данной архитектуры.

В соответствии с поставленными целями были сформулированы следующие задачи:

1. Разработать безопасную архитектуру.
2. При помощи программы Oracle VirtualBox установить Oracle Database Firewall и настроить его.
3. Создать на сервере MySQL Server базу данных и заполнить её.
4. Произвести серию атак на базу данных, используя SQL Power Injector.

Построение безопасной архитектуры

Теоретическая часть

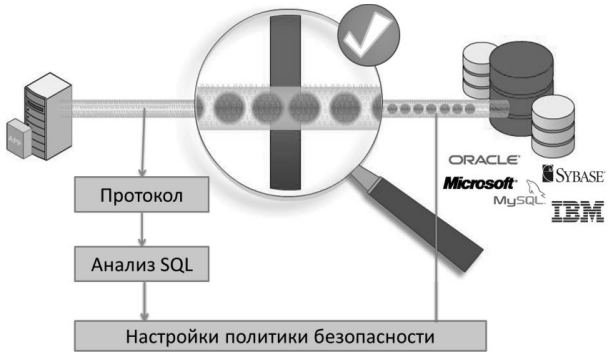
Задачи:

1. Мониторинг трафика и исключение неавторизованного доступа к базам данных, исключение SQL инъекций, позволяющих несанкционированно повышать привилегии и получать доступ к конфиденциальным данным.
2. Аккуратный грамматический анализ SQL выражений.
3. Высокая масштабируемость и производительность.
4. Возможность реализации политик «белого» и «чёрного» списков.

Проектирование:

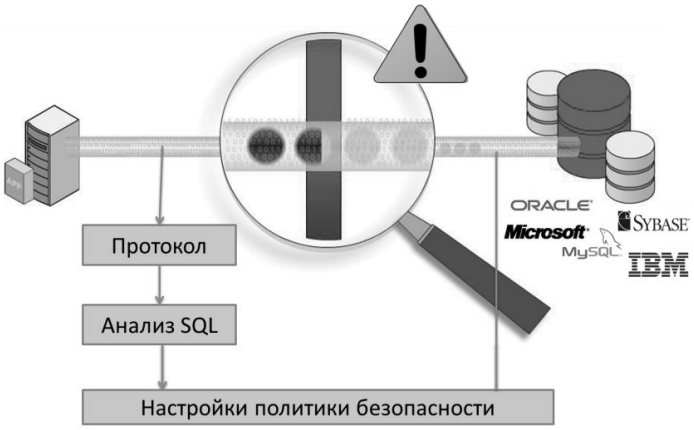
Всем данным задачам соответствует только продукт Oracle — Oracle Database Firewall, при помощи которого мы и будем обеспечивать безопасность. В качестве базы данных выбрана MySQL, так как она проста в установке и эксплуатации, в отличие от Oracle DB 11 и IBM DB 2.

1. Мы соединяем рабочее приложение с базой данных и параллельно к ним подключаем фаервол, который будет анализировать SQL запросы...

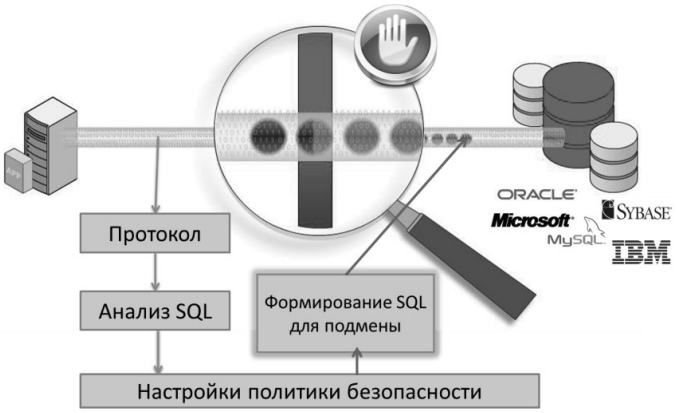


2. ...и сверять их с политиками безопасности.

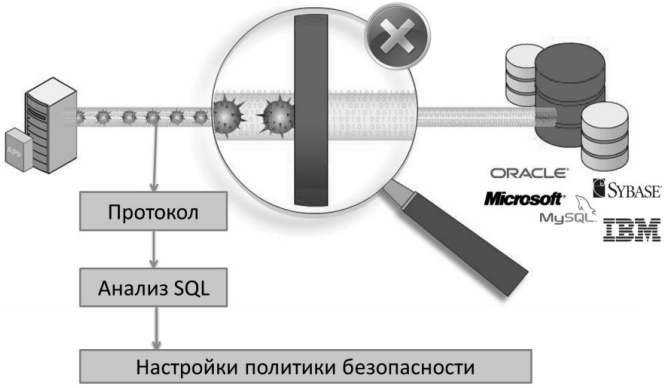
3. В случае несоответствия запроса с политиками безопасности включается механизм блокирования запроса...



4. ...и формирования SQL для подмены запроса.



5. По завершении работы механизма блокировки, вся работа приложения блокируется путём отправления терминальных сигналов.



Практическая часть

Установка Oracle VirtualBox

1. Загружаем последнюю версию Oracle VirtualBox с сайта www.virtualbox.org.
2. Запускаем установочный файл.
3. Следуем инструкциям установки.

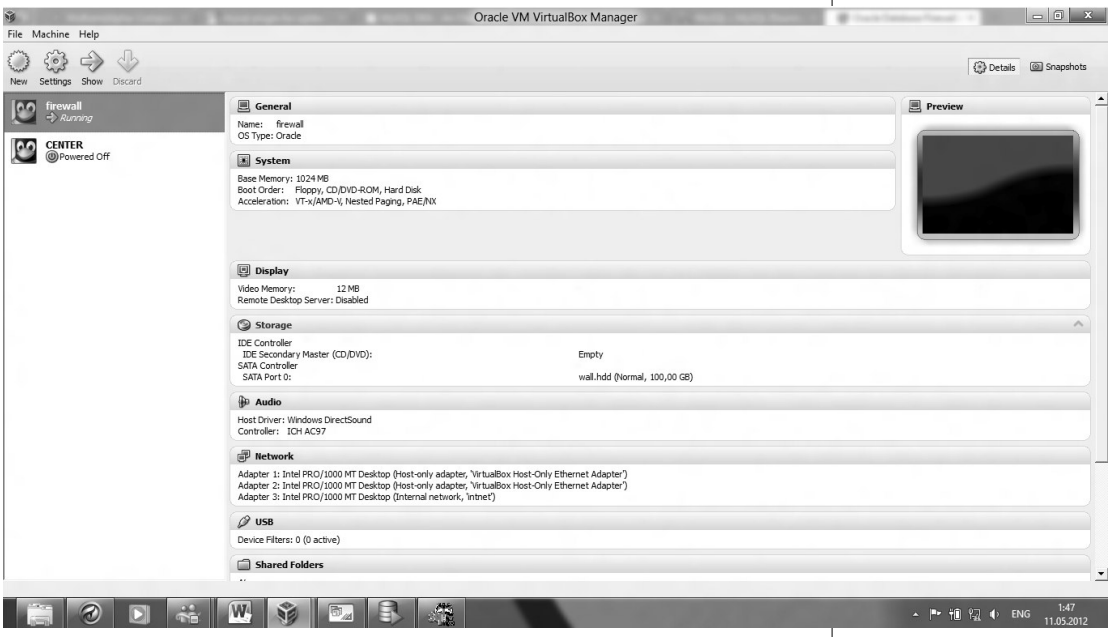


Рис. 1. Главный экран Oracle VirtualBox

Установка Oracle Database Firewal

Создаём виртуальную машину: операционная система — Linux, Oracle 32-bit.

1. Выделяем оперативную память — 1024mb.
2. Создаём динамически расширяемый HDD размером 100 Gb.
3. Создаём 3 сетевых интерфейса. 1-й, 2-й Host Only, 3-й Internal.
4. В виртуальный дисковод «вставляем» disk_1 и далее (по подсказкам системы установки).
5. В итоге имеем машину с установленным DBFW. На синем экране (в тексте — чёрный) указываем IP-адрес management port_a (например, 192.168.56.30, 255.255.255.0, Gateway: 192.168.56.254).
6. С Host-машины (сетевому интерфейсу которой присваиваем адрес 192.168.56.10) браузером «стучимся» по https к: 192.168.56.30, соглашаемся с представленным сертификатом и входим в панель управления admin/admin. Это подтверждает, что DBFW установился правильно.

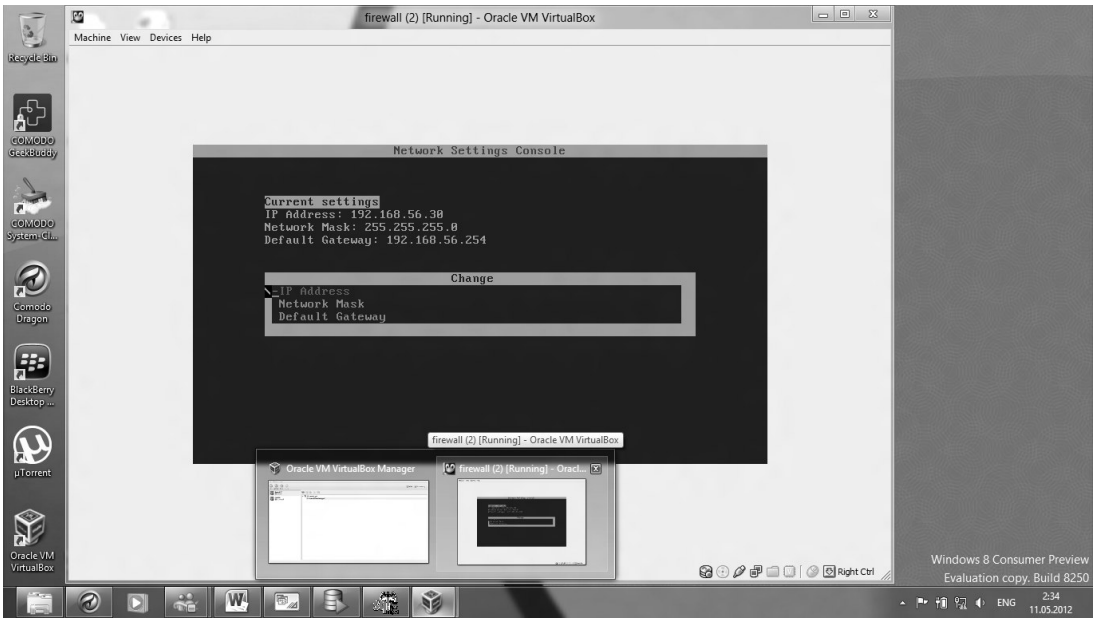


Рис. 2. Экран настройки соединения Oracle Database Firewall.

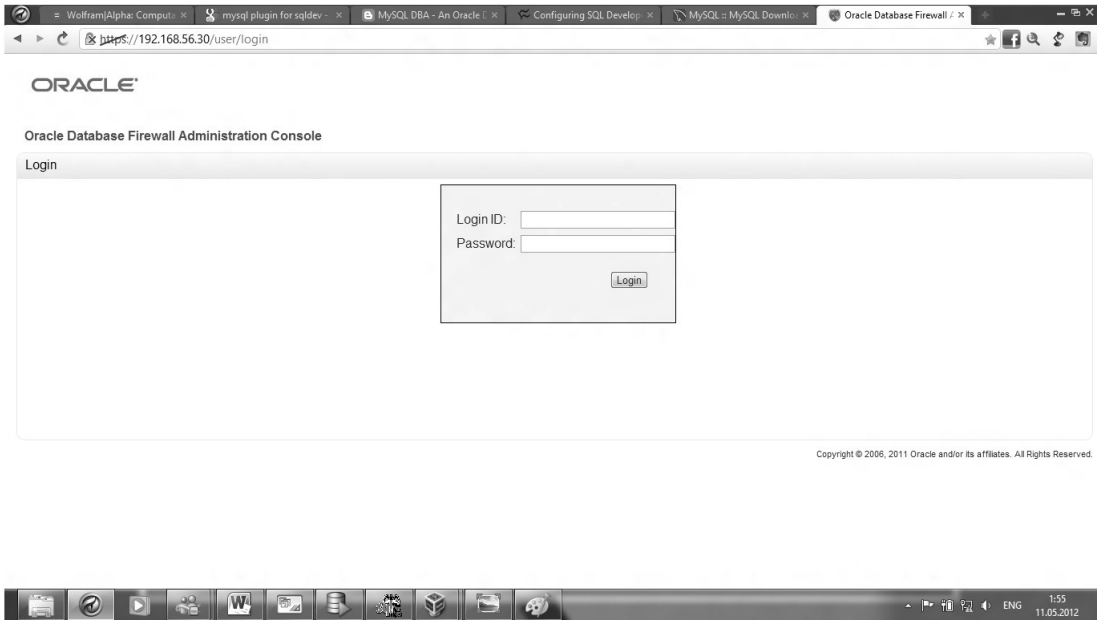


Рис. 3. Login экран Oracle Database Firewall

Создание базы данных на MySQL Server:

1. Загружаем с сайта www.mysql.org MySQL Installer.
2. Запускаем установочный файл.
3. Устанавливаем полную версию и следуем дальнейшим инструкциям установщика.
4. Запускаем MySQL Workbench (рис.4).
5. Заходим на свой сервер MySQL_5.5 и проверяем, что он работает (рис.5).
6. Выбираем в панели раздел Database и нажимаем «Query Database».
7. В открывшемся окне выбираем стандартное подключение и вводим свой пароль.
8. Видим открывшееся окно редактора (рис.6).
9. Следуем простым шагам по созданию и заполнению баз данных в SQL.

— Сначала взглянем на список уже имеющихся баз данных (вводим SHOW DATABASES).

— Видим, что есть 2 базы данных — mysql и test.

— Создадим свою базу данных (*tester*) (вводим CREATE DATABASE tester).

— Задаём нашу базу данных как текущую (вводим USE tester).

— Создаём таблицу *users* с полями *name, email, job* типа VARCHAR (вводим CREATE TABLE users (name VARCHAR(128), email VARCHAR(128), job VARCHAR(128)).

— Вставляем в неё одну запись *John Smith, john@tester.com, DBA* (вводим INSERT INTO users VALUES('John Smith', 'john@tester.com', 'DBA')).

— База данных готова.

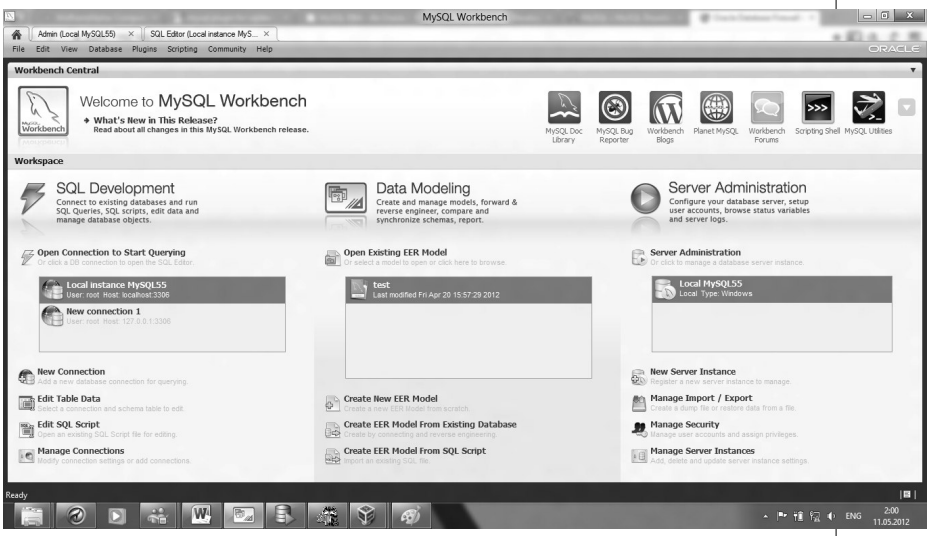


Рис. 4. MySQL Workbench

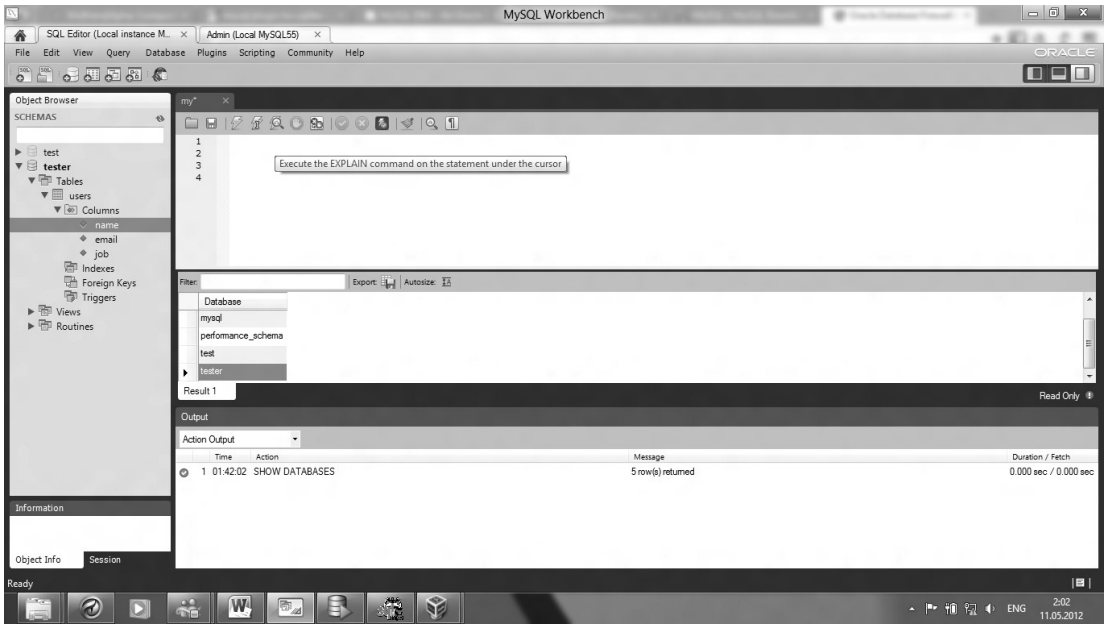


Рис.5. Экран состояния сервера

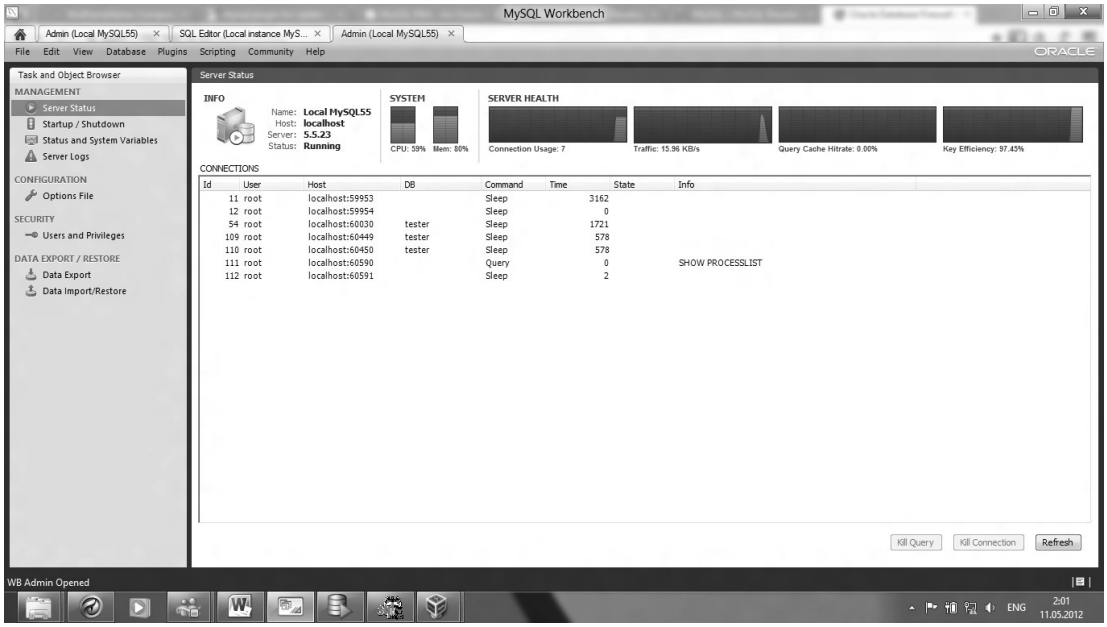


Рис. 6. Окно SQL редактора

Настройка Oracle Database Firewall:

1. Запускаем фаервол.
2. Вводим в окне свои входные данные.
3. Попадаем на рабочий экран Oracle Database Firewall (рис. 7).
4. Выбираем пункт меню Manage Databases.
5. Следуем инструкциям регистратора баз данных.
6. Фаервол подключён к базе данных.

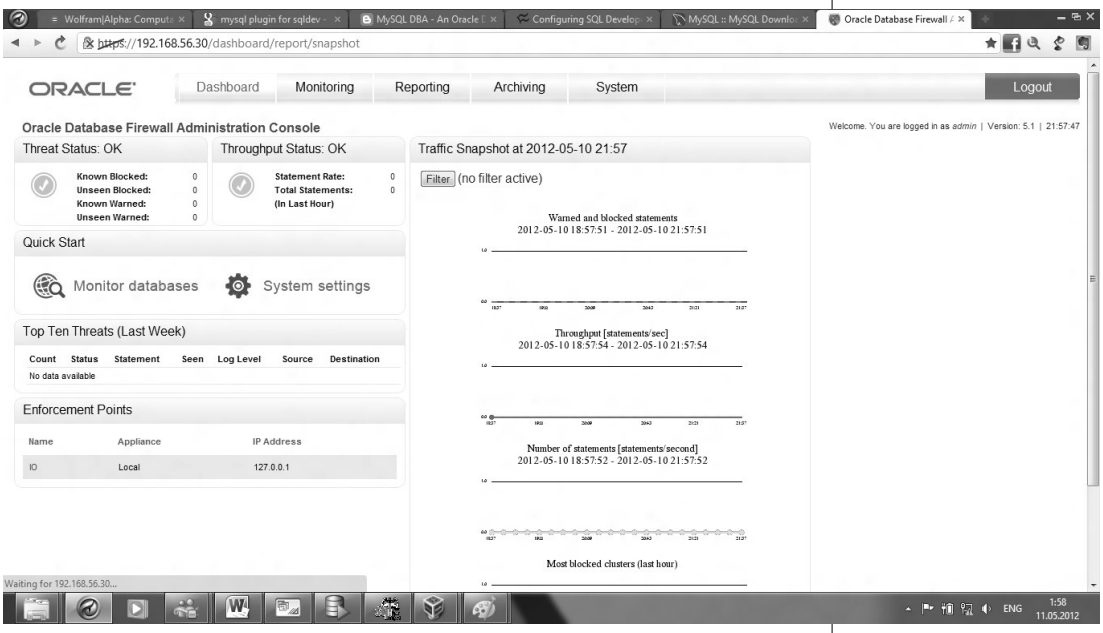


Рис. 7. Рабочий экран Oracle Database Firewall

Проверка работы безопасной архитектуры:

1. Устанавливаем программу SQL Power Injector.
2. Запускаем её.
3. В графе *Enter URL* вводим *localhost:3306* (место расположения базы данных и номер порта).
4. Нажимаем *Start*.
5. Если фаервол блокировал инъекцию, то в нижней графе появится сообщение об ошибке (рис. 8).

Выводы

В результате проведённой нами работы можно сделать следующие выводы:

1. Разработанная нами архитектура способна обеспечивать безопасность баз данных.
2. Она легко разворачивается и настраивается на любой платформе.
3. Она предоставляет нам аккуратный грамматический анализ SQL выражений.

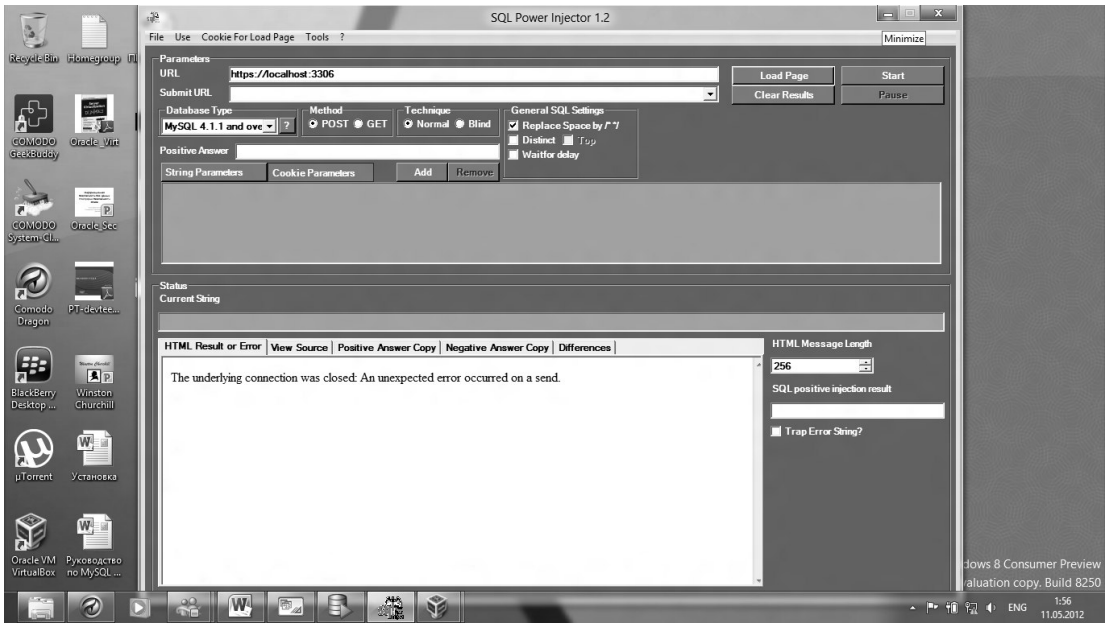


Рис. 8. Интерфейс программы SQL Power Injector

4. Данная архитектура может реализовать технологии «белого» и «чёрного» листа.

5. С её помощью возможно исключение неавторизованного доступа к базам данных, исключение SQL инъекций, позволяющих несанкционированно повышать привилегии и получать доступ к конфиденциальным данным.