Математические структуры и моделирование 2018. № 4(48). С. 147–153

УДК 004.056
DOI: 10.25513/2222-8772.2018.4.147-153

ИМПОРТ И ЭКСПОРТ РОЛЕВОЙ ПОЛИТИКИ БЕЗОПАСНОСТИ В СУБД ORACLE

Ю.С. Ракицкий
к.т.н., доцент, e-mail: yrakitsky@gmail.com

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. В статье рассматривается ролевая политика безопасности, реализованная в СУБД Oracle. Модель ролей в СУБД может быть достаточно сложной, содержать большое количество ролей и полномочий. Эта информация хранится в большом количестве связанных между собой таблиц. Реализован механизм извлечения данных, сопоставленных ролевой политике безопасности в СУБД Oracle, и представление этой информации в формате GraphML. Также реализована возможность импорта ролевой политики безопасности, представленной в формате GraphML, в СУБД Oracle.

Ключевые слова: ролевая модель, СУБД, язык представления графов.

Введение

Современные базы данных используют ролевую модель для выдачи полномочий пользователям, выполняя таким образом ролевое разграничение доступа. При использовании ролевой политики безопасности задаётся множество разрешённых системных операций путём введения дополнительных объектов – ролей, наделённых набором разрешённых доступов.

Часто возникает необходимость проанализировать модель, построенную средствами СУБД с теоретической точки зрения, так как для больших систем управление огромным количеством ролей, пользователей и разрешений является сложной задачей, которую трудно выполнять малой группой администраторов безопасности.

Сложность администрирования безопасности СУБД связана с тем, что в СУБД достаточно трудно получить общую картину о политике безопасности, так как информация находится в большом количестве взаимосвязанных друг с другом таблиц [1]. Для решения этой проблемы отношения между субъектами удобно представить в каком-либо формате, пригодном для последующего анализа.

Оценка реальной модели с теоретической точки зрения позволяет выявить ошибки администрирования, проанализировать отношения субъектов и объектов, а также проверить адекватность построенной политики безопасности [2].

Помимо непосредственного анализа модели возможно проведение ряда работ по улучшению текущей ролевой модели [3], уменьшению количества ролей,

выявлению и устранению уязвимостей. Также после анализа и оптимизации политики безопасности нужно внедрить новые правила, поэтому удобно автоматизировать применение правил из-за их большого количества и возможных ошибок администратора базы данных при непосредственной настройке.

В связи с этим появляется необходимость в экспорте ролевой политики безопасности из базы данных в каком-либо формате для последующего анализа. Цели, которые преследуются в ходе анализа, могут быть очень разнообразны. Например, в работе [4] описан процесс автоматизации сортировки полномочий по значению риска утечки. А в работе [5] описаны способы выполнения эквивалентных преобразований иерархии ролей для оптимизации RP-орграфа или приведения его к древовидной структуре. Именно для осуществления подобного рода задач и необходимо данное программное решение. Так как в базе данных насчитываются сотни ролей и полномочий, наличие инструмента, позволяющего автоматически извлечь нужную информацию, существенно упростит дальнейшие действия для администратора.

1. Основные источники информации и формат представления данных

Таблица DBA_ROLES в СУБД Oracle содержит информацию обо всех ролях, которые имеются в базе данных, включая системные предопределённые роли.

В рамках тестирования производилась выборка одного столбца из данной таблицы – $ROLE$.

Таблица $ROLE\_TAB\_PRIVS$ включает в себя информацию обо всех привилегиях, выданных ролям, имеющимся в системе. Выборка производится для каждой роли с условием $WHEREROLE = ROLE\_NAME$. Необходимые для работы столбцы таблицы: $TABLE\_NAME, PRIVILEGE, OWNER$.

В рамках исследований были выделены 16 объектных привилегий в СУБД Oracle, для хранения информации о наличии или отсутствии соответствующей привилегии достаточно одного бита, поэтому можно представлять роль как битовую строку длины 16. Список привилегий можно при необходимости расширить путём внесения нового соответствия. Перечень объектных привилегий и соответствующие позиции битов в битовой строке представлены в таблице 1.

В качестве примера приведём роль $TEST\_ROLE$. Указанная роль имеет права на удаление записей, добавление записей, обновление записей и выборку из объекта $TEST\_TABLE$. Поэтому битовая строка прав $TEST\_ROLE$ на $TEST\_TABLE$ будет выглядеть следующим образом: 0100001000010100.

2. Алгоритмы импорта и экспорта данных

Выделим основные этапы, выполнение которых необходимо для осуществления экспорта данных из СУБД Oracle:

1. Поиск всех ролей в базе данных.

Таблица 1. Заданная нумерация битов для объектных привилегий

Номер бита	Объектная привилегия
0	ALTER
1	DELETE
2	EXECUTE
3	DEBUG
4	FLASHBACK
5	INDEX
6	INSERT
7	ON COMMIT REFRESH
8	QUERY REWRITE
9	READ
10	REFERENCES
11	SELECT
12	UNDER
13	UPDATE
14	WRITE
15	DEQUEUE

2. Для каждой роли запуск процедуры поиска выданных привилегий на объекты системы.
3. Сохранение роли и списка её привилегий на объекты.
4. Заполнение графа с помощью Graph API от TinkerPop.
5. Запись графа в выбранном формате.

Аналогичным образом, выделим основные этапы, выполнение которых необходимо для осуществления импорта данных из СУБД Oracle:

1. Считывание графа из файла заданного формата.
2. Запуск процедуры для считывания данных из графа с помощью Graph API.
3. Для каждой роли запуск процедуры для выдачи и отзыва привилегий на объекты.

3. Компьютерный эксперимент

Программный продукт представляет собой консольное приложение для взаимодействия с оператором, реализованное на языке программирования Java.

Основное меню программы содержит следующие пункты:

1. Установить соединение.
2. Считать данные из БД.
3. Записать данные в БД.
4. Выйти.

Пункт меню «Установить соединение» предлагает пользователю ввести ряд параметров для подключения к базе данных:

- Имя хоста / IP-адрес сервера.
- Порт.
- SID.
- Пароль пользователя SYS.

Осуществление действий, производимых программой, возможно только с привилегиями пользователя SYS. Подключение к базе данных производится с административной привилегией SYSDBA: «SYS AS SYSDBA». В случае успешного подключения появится надпись «Соединение установлено», иначе выведется предупреждение «Введены некорректные данные или хост недоступен».

Пункт «Считать данные из БД» предлагает выбрать формат файла (GraphML или GraphSon), затем ввести путь к файлу для сохранения ролевой политики.

Пункт «Записать данные в БД» предлагает выбрать формат файла (GraphML или GraphSon), затем ввести путь к файлу для считывания из него ролевой политики.

Для выполнения действий по считыванию и записи данных в базу данных необходимо предварительно установить соединение с базой данных.

Пункт «Выйти» предназначен для завершения работы программы.

После выполнения каждого из пунктов, кроме выхода, вызывается функция по очистке консольного окна, и меню печатается снова.

4. Тестирование

Тестирование осуществлялось как с точки зрения производительности, так и корректности работы. Аппаратное и программное окружение, на котором осуществлялось тестирование:

1. Процессор — Intel Core i5-3230M 2.60 GHz.
2. ОЗУ – 8.00 Гб.
3. Windows 10 x64.
4. ORACLE 11g.
5. Java version 1.8.0_51.

Тестирование производительности проводилось с помощью функции System.nanoTime() путём измерения начального времени (перед запуском) и конечного времени (после запуска) и нахождения разницы между этими значениями.

Из таблицы 2 можно заметить, что среднее значение считывания данных из базы данных для формата GraphML составляет около 1100 миллисекунд, в то время как среднее значение для формата GraphSon составляет примерно 1300 миллисекунд. Данную разницу можно объяснить накладными издержками при сохранении файлов в разных форматах. В целом можно отметить, что данная операция выполняется довольно быстро.

Из таблицы 3 можно увидеть, что среднее значение записи данных в базу данных для формата GraphML составляет около 169 секунд, а среднее значение для формата GraphSon составляет примерно 170 секунд. Разница между

Таблица 2. Производительность чтения из базы данных

Номер теста	GraphML, мс	GraphSon, мс
1	999	1293
2	1059	1261
3	1068	1274
4	1064	1278
5	1052	1262
6	1049	1270

Таблица 3. Производительность записи в базу данных

Номер теста	GraphML, сек.	GraphSon, сек.
1	167	169
2	170	171
3	172	175
4	171	172
5	166	166
6	169	170

форматами невелика, её можно объяснить разными накладными расходами на обработку.

Стоит отметить, что запись происходит значительно дольше считывания из-за следующих факторов:

1. Необходимо выполнять формирование запросов на выдачу прав.
2. Необходимо явно отзывать права, для которых в соответствие поставлено отсутствие данного права. Из-за чего в разы увеличивается количество операций.
3. Чтобы выдать или отозвать права, создаётся курсор. Из-за их большого количества также происходят временные задержки.

Результаты зависят от того, насколько нагружена система, так как запись данных в базу данных является очень трудоёмкой операцией, и наличие других ресурсоёмких процессов может серьёзно замедлить выполнение программы.

В целом данные результаты довольно условны, так как всё зависит не только от производственных мощностей, но и от количества объектов в исследуемой базе данных и связей между ними. Но вместе с тем можно сделать вывод, что программа выполняет свои задачи в достаточно приемлемое время, которое не требует от пользователя предварительного планирования и длительного ожидания.

Тестирование корректности чтения данных производилось с помощью создания объектов в базе данных и установления различных связей между ними, затем осуществлялось считывание ролевой политики безопасности в файл формата GraphML. Тестирование корректности записи данных происходило путём изменения файла формата GraphML, внесения изменений с помощью этого

файла и проверки записей в базе данных на соответствие внесённым.

Тестовая конфигурация:

1. Роли R1, R2, R3.
2. Таблицы O1, O2, O3.
3. R1 имеет права ALTER, DELETE, INSERT на таблицу O1.
4. R1 имеет права SELECT и UPDATE на таблицу O2.
5. R2 имеет права DELETE и INSERT на таблицу O3.
6. R3 имеет право SELECT на таблицу O2.
7. R3 имеет права SELECT, INSERT, DELETE на таблицу O3.

Данная конфигурация была настроена в тестовой базе данных. В результате операции считывания получен файл, содержимое файла полностью соответствует настройкам, осуществлённым в базе данных.

В полученном файле было произведено следующее изменение данных: ребро с идентификатором 10 поменяло значение на 0000001000010000, ребро с идентификатором 4 — на 0100000000000100, ребро с идентификатором 5 — на 1100001000000100, ребро с идентификатором 7 -– на 0000001000000000, ребро с идентификатором 9 — на 0100000000010000. Проверка осуществлялась для каждой из ролей. Для роли R1 использовалась команда «SELECT TABLE_NAME, PRIVILEGE FROM ROLE_TAB_PRIVS WHERE ROLE = 'R1'». Результаты обработки отображены в таблице 4.

Таблица 4. Привилегии роли R1

TABLE NAME	PRIVILEGE
O2	UPDATE
O1	UPDATE
O1	INSERT
O2	DELETE
O1	DELETE
O1	ALTER

Для роли R2 получена одна привилегия — INSERT на таблицу O3.

Привилегии, полученные для роли R3, отображены в таблице 5.

Таблица 5. Привилегии роли R3

TABLE NAME	PRIVILEGE
O2	DELETE
O3	INSERT
O3	SELECT
O2	SELECT

Полученные в ходе тестирования результаты являются корректными, что позволяет сделать вывод о том, что запись данных в базу данных производится правильно и соответствует ожиданиям пользователя.

Заключение

Предложенный подход к экспорту и импорту ролевой политики безопасности может применяться для широкого класса задач, являясь при этом вспомогательным инструментом для их осуществления. Использование предложенного инструмента возможно для осуществления анализа и оптимизации ролевой политики безопасности, а также для поиска потенциальных уязвимостей.

В результате тестирования было выявлено, что приложение работает достаточно быстро. Также была проведена проверка корректности работы программы, которая дала положительные результаты.

Литература

1. Database Documentation - Oracle Database // Oracle Help Center. URL: `https://docs.oracle.com/en/database` (дата обращения: 25.01.2017).
2. Гайдамакин Н.А. Теоретические основы компьютерной безопасности: учебное пособие. Екатеринбург : изд-во Урал. ун-та, 2008. 212 с.
3. Девянин П.Н. Модели безопасности компьютерных систем. М. : Издательский центр «Академия», 2005. 144 с.
4. Белим С.В., Богаченко Н.Ф. Применение метода анализа иерархий для оценки рисков утечки полномочий в системах с ролевым разграничением доступа // Информационно-управляющие системы. 2013. № 4. С. 67–72.
5. Белим С.В., Белим С.Ю., Богаченко Н.Ф. Теоретико-графовый анализ ролевой политики безопасности // Математические структуры и моделирование. 2009. № 19. С. 85–96.

IMPORT AND EXPORT OF ROLE-BASED SECURITY POLICIES IN ORACLE DBMS

Y.S. Rakitskiy
Ph.D. (Eng.), Associate Professor, e-mail: yrakitsky@gmail.com

Dostoevsky Omsk State University, Omsk, Russia

Abstract. The article discusses the role-based security policy implemented in the Oracle DBMS. The role model in a DBMS can be quite complex, contain a large number of roles and authorities. This information is stored in a large number of related tables. A mechanism has been implemented to extract data mapped to role-based security policies in an Oracle DBMS and to present this information in GraphML format. It also implemented the ability to import role-based security policy, presented in GraphML format, in Oracle DBMS.

Keywords: role-based model, DBMS, graph representation language.

Дата поступления в редакцию: 20.11.2018