

УДК 004.021

А.С. Марковский, канд. техн. наук, ст. науч. сотрудник, Военно-космическая академия им. А.Ф. Можайского (Санкт-Петербург, Россия) (e-mail: lexh26@list.ru)

Н.И. Свеколкин, ст. науч. сотрудник, Военно-космическая академия им. А.Ф. Можайского (Санкт-Петербург, Россия) (e-mail: ins_61@mail.ru)

МЕТОД ОБНАРУЖЕНИЯ АНОМАЛИЙ SQL-ЗАПРОСОВ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

В условиях ужесточения требований в сфере безопасности информации (условий ее сохранности), труднореализуемых на практике, растущего числа внешних дестабилизирующих факторов (в том числе высокого уровня ложных тревог), увеличения объемов и скорости изменения информации, и присущих большинству баз данных недостатков, высока вероятность возникновения аномалий в процессе эксплуатации (сбора, обработки и хранения) реляционных баз данных.

В статье подробно описывается метод построения формальной грамматики, выполняемого SQL-запроса реляционных баз данных. Данный подход рассматривает формальную грамматику, с математической точки зрения как модель, задающую множество дискретных объектов в виде описания изначальных объектов и правил построения новых объектов из исходных и уже созданных. Таким образом, формируется система правил для дальнейшей работы, представленная в виде системы уравнений. Описываемый способ позволяет определить математические свойства инвариантов подобия выполняемых SQL-запросов реляционных баз данных, предназначенных для сбора, хранения и анализа статистических данных, таких, как контрольные данные о работе программных и аппаратных средств, различных статистических сведений о населении, данных о производстве продукции и т.д.

Полученные в ходе экспериментальной реализации результаты тестирования демонстрационного прототипа системы обнаружения аномалий, реализованного на основе предложенного метода, представлены в сравнении с некоторыми существующими и применяемыми системами защиты. Предлагаемое в статье решение является эффективным, простым и универсальным для большинства применяемых в настоящее время реляционных баз данных, кроме того, обладает низкой себестоимостью финансовых издержек при практической реализации.

Ключевые слова: SQL-запрос, инвариант подобия, реляционная база данных.

DOI: 10.21869/2223-1560-2018-22-1-53-61

Ссылка для цитирования: Марковский А.С., Свеколкин Н.И. Метод обнаружения аномалий SQL-запросов реляционных баз данных // Известия Юго-Западного государственного университета. 2018. Т. 22, № 1(76). С. 53-61.

Введение

Задача обнаружения аномалий выполняемых SQL-запросов в условиях внешних дестабилизирующих факторов (различных возмущающих воздействий, сбоев и т.д.) функционирования баз данных (далее – БД) является одной из задач защиты информации от несанкционированного доступа. Сложность решения названной задачи заключается в том, что проектируемая система обнаружения аномалий не должна содержать в себе точную копию контролируемого SQL-запроса. В противном случае будет эко-

номически эффективнее произвести дублирование SQL-запроса и выполнять обнаружение и коррекцию аномалий SQL-запросов на основе точной копии.

В условиях создания организациями собственных реляционных БД, стремительного увеличения объемов быстро изменяемой информации с постоянно увеличивающейся долей неструктурированных данных, возникает острая необходимость развития механизмов по их контролю и диагностике.

В процессе работы с БД происходят:

1) сбор и согласование данных из гетерогенных источников данных;

2) предварительный анализ данных (выявление штатных и внештатных ситуаций);

3) получение агрегированных значений и статистических оценок за нужные периоды (на каждом временном срезе);

4) удаление лишней информации (не имеющей значения для задачи анализа);

5) сжатие и восстановление данных (отдельных полей таблиц, таблиц, БД);

6) ведение регламентов, описывающих расписание и схему работы процедур сжатия, агрегации и предварительного анализа;

7) представление данных в удобном виде (отображение данных с возможностью выбора периода, объекта, изделия, сравнение данных за разные периоды).

Кроме того существует также вероятность возникновения аномалий в процессе хранения данных. Его можно рассматривать как состоящий из трех частей: организации физической памяти (оптические диски, флеш-память, жесткие диски, технология Redundant array of independent disks), логической организации данных (хранилища данных, применяемые технологии) и сжатия данных (методы сжатия и архивации). И на каждом из них также существует возможность появления аномалий.

Описание способа решения задачи

Для решения подобного рода задач нужно разработать формальную грамматику инвариантов подобия (далее – ИП), позволяющую первоначально обозначить математические свойства ИП выполняемых SQL-запросов, а затем сформулировать строгие утверждения, необходимые для контроля структурно-функциональной целостности реализуемых SQL-запросов БД.

Вопросам обнаружения аномалий в SQL-запросах посвящены работы таких авторов, как Д.Вигна, Ф.Валер, А.Карма, Д.Мутц, П.Вонг, Э.Бернито, К.Чанг, С.Носок, А.Павлов, А.Спалки, В.Лоу, П.Зегжды, Герасименко В.А. и многих других. Описываемые в указанных работах приемы обнаружения аномалий в SQL-запросах БД предлагают оценивать запрос до его выполнения на основе различных синтаксических характеристик текста запроса. При таком способе рассматривается статичная модель SQL-запроса, не учитывающая текущее состояние хранящейся в БД информации (данных).

Предлагается рассматривать формальную грамматику как математическую модель, задающую множество дискретных объектов в виде описания изначальных объектов и правил построения новых объектов из исходных и уже созданных. Под объектами, в нашем случае, будем понимать символические представления ИП. Всякая грамматика начинается с указания алфавита, т.е. набора символов, при помощи которого строятся конструкции языка.

Синтаксис формального языка определяется некой совокупностью правил (порождающей системой), формирующей из небольшого набора исходных конструкций все их возможные комбинации, т.е. язык является множеством разрешенных правилами сочетаний исходных вариантов. Помимо этого, синтаксис содержит формулировку условия, реализуемую для итоговых конструкций языка и не выполняющуюся в противном случае.

А кроме синтаксиса задается система правил, предоставляющих конструкциям языка придать смысл – эти правила образуют семантику языка. Таким образом, формальная грамматика – это система

правил, описывающая множество конечных последовательностей символов формального алфавита. Конечные цепочки символов называются предложениями формального языка, а само множество цепочек – языком, описываемым данной грамматикой.

Для построения формальной грамматики ИП используем следующий метод:

- Определить терминальный алфавит (ИП).
- Преобразовать значения реализаций SQL-запросов в исходные данные (терминальный алфавит).
- Сформировать предложения (связи между параметрами).
- Определить свойства формальной грамматики ИП, необходимые для контроля структурно-функциональной целостности SQL-запросов.

На первом этапе необходимо сформировать терминальный алфавит (ИП).

В операторах языка SQL можно выделить 2 категории размерностей (ИП):

- конструкции условий;
- конструкции согласований.

Пусть задано множество контрольных точек Π_i . При формировании терминального алфавита i -го инварианта размерности (V_{Ti}) выполняемого SQL-запроса БД необходимо построить отображение:

$$W_T : (\Pi_i) \rightarrow V_{Ti}.$$

В ряде случаев, SQL-запрос содержит обращение (ряд параметров) к хранимой процедуре, в этом случае отображение W_T примет следующий вид:

$$W_T : (\Pi_i, P_i) \rightarrow V_{Ti},$$

где P_i – некое множество вычисляемых хранимых процедур.

На этапе преобразования значения реализаций SQL-запросов в исходные данные (терминальный алфавит) необходимо провести преобразование i -го критерия $\Pi_i \in A$ в символы терминального алфавита V_{Ti} , полученного при реализации отображения W_T .

Данные преобразования определяются отображением:

$$W_{\Pi} : (A_i, V_{Ti}) \rightarrow \Pi_{Ti},$$

где $\Pi_{Ti} \subseteq V_{Ti}$.

Соответственно для случая обращения к хранимым процедурам:

$$W_{\Pi} : (A_i, P_i, V_{Ti}) \rightarrow \Pi_{Ti}$$

в общем случае отображение $W_{\Pi} = W'_{\Pi} \cdot W''_{\Pi}$, где $W'_{\Pi} : (A_i, P_i) \rightarrow H_{\Pi i}$,

$$W''_{\Pi} : (H_{\Pi i}, V_{Ti}) \rightarrow \Pi_{Ti},$$

где $H_{\Pi i}$ – множество значений вычисляемых функций.

На третьем этапе необходимо сформировать предложения (связи между параметрами). Введем множество различных критериев $\Pi_T = \{\Pi_{Ti}\}$. Формирование требуемых последовательностей (правил) может быть получено в результате отображения:

$$W_P : \Pi_T \rightarrow V_T^*,$$

где V_T^* – свободная полугруппа на терминальном алфавите $V_T = \bigcup_{i=1}^n V_{Ti}$;

n – общее количество критериев (ИП).

Для реализации отображения W_P введем ряд операций на множестве Π_T .

Операция конкатенации. Для критериев $\Pi_1, \Pi_2 \in \Pi_T$ определим ассоциативную операцию конкатенации, формулирующую цепочку символов $W = \Pi_1 \Pi_2$.

Расширяя действия данной операции на цепочки символов W_1 и W_2 , получим $W = W_1W_2$.

Данная операция позволяет получать различные цепочки символов терминального алфавита, соответствующие наборам значений измеряемых параметров.

Операция смещения. Смещением цепочек $W_1 = a_1a_2...a_n$ и $W_2 = b_1b_2...b_n$, $W_1 \in L_1$, $W_2 \in L_2$, будем называть цепочку $W = a_1b_1a_2b_2...a_nb_n$, получаемого в результате отображения:

$$\alpha_1 : (W_1, W_2) \rightarrow W.$$

Данная операция может быть использована при формировании единой цепочки символов.

Введем следующее утверждение: класс регулярных языков замкнут относительно операции смещения.

Доказательство. Пусть цепочки $W_1 = a_1a_2...a_n$ и $W_2 = b_1b_2...b_n$ принадлежат регулярным языкам L_1 и L_2 , с грамматиками $G_1 = (N_1, T_1, R_1, S_1)$ и $G_2 = (N_2, T_2, R_2, S_2)$.

Пусть так же для формирования элементов $a_i, b_i, a_{i+1}, b_{i+1}$ на множествах P_1 и P_2 , описывающих инварианты 1-го рода, используются правила:

$$A_i \rightarrow a_i A_{i+1},$$

$$C_i \rightarrow b_i C_{i+1},$$

$$A_{i+1} \rightarrow a_{i+1} A_{i+2},$$

$$C_{i+1} \rightarrow b_{i+1} C_{i+2}.$$

Рассмотрим грамматику G_3 , реализующую операцию смещения:

$$G_3 = (N_3, T_3, R_3, S_3),$$

$$N_3 = N_1 \cup N_2,$$

$$T_3 = T_1 \cup T_2,$$

$$S_3 = S_1 \cup S_2.$$

Множество R_3 формируется из множеств R_1 и R_2 , описывающих инварианты 2-го рода, в результате следующих преобразований:

$$A_i \rightarrow a_i C_i,$$

$$C_i \rightarrow b_i A_{i+1},$$

$$A_{i+1} \rightarrow a_{i+1} C_{i+1},$$

$$C_{i+1} \rightarrow b_{i+1} A_{i+2}.$$

В результате грамматика G_3 порождает язык с цепочками $W_3 = a_1b_1a_2b_2...a_nb_n$.

Операция сжатия. Сопоставим каждому символу $a \in T$ алфавит T_a и множество $\tau(a) \subseteq T_a^*$, получаемые посредством отображения τ .

Пусть так же $\tau(a_1...a_n) = \tau(a_1)... \tau(a_n)$ для каждой цепочки $a_1...a_n \in T^*$, тогда отображение примет вид:

$$\tau : 2^{\left(\bigcup_a T_a\right)^*} \rightarrow T^*.$$

В данном случае построение цепочки $W' = W_1aW_3, a \in T$ по цепочке $W = W_1W_2W_3$ возможно и при пустых цепочках W_1 и (или) W_2 . Операция τ позволяет объединять результаты анализа.

Операция подстановки. Является обратной к операции сжатия:

$$\nu : T^* \rightarrow 2^{\left(\bigcup_a T_a\right)^*}.$$

Существенно, что определение рассмотренных операций в совокупности с общепринятыми операциями (объединением, пересечением, дополнением, транспозицией и итерацией) позволило сформировать из значений анализируемых ИП SQL-запросов (контрольных точек) – информационные предложения (цепочки ИП SQL-запросов), являющие собой эталоны структурно-функциональ-

ной целостности SQL-запросов реляционных БД.

На последнем этапе нам нужно определить свойства формальной грамматики ИП, применяемые для контроля структурно-функциональной целостности SQL-запросов.

Необходимо учитывать, что терминальный словарь грамматики должен состоять из всех символов, применяемых при формировании цепочек, составляющих заданное множество, а результатом решения задачи должны стать нетерминальный словарь и схема грамматики. Создание обозначенных объектов является весьма сложным, так как осуществляется неформально, и требует рассмотрения всевозможных вариантов построения цепочек заданного множества и синтеза правил их формирования. Построение осложняется еще и тем, что подобно любой задаче синтеза, имеет множество вариантов реализации.

Возможность построения цепочек основывается на базе правил вывода грамматики, фундаментом для формирования которых служит возможность выделения структуры заданного множества цепочек. Данный прием предусматривает расчленение цепочек, составляющих заданное множество, на части, позволяя установить повторяющиеся части цепочек и части, входящие во все цепочки в неизменном виде. Применение расчленения на части дает возможность увидеть структуры цепочек заданного множества.

Для каждого выявленного элемента структуры введем обозначения. Множество подобных обозначений формирует основу словаря нетерминальных символов некоторой грамматики. Следующим шагом построения выступает обнаружение последовательностей в элементах

структуры, которые могут содержаться в заданных цепочках. Данные последовательности выступают основой формирования правил грамматики.

В общем случае, если описано множество цепочек, представляющих собой некоторый язык, и требуется построить грамматику, порождающую это множество цепочек, то применим следующий способ:

1. Выписать несколько примеров из заданного множества цепочек.
2. Проанализировать структуру цепочек, выделяя начало, конец, повторяющиеся символы или группы символов.
3. Ввести обозначения для сложных структур, состоящих из групп символов. Такие обозначения являются нетерминальными символами искомой грамматики.
4. Построить правила для каждой из выделенных структур, используя для задания повторяющихся структур рекурсивные правила.
5. Объединить все правила.
6. Проверить с помощью выводов возможность получения цепочек с разной структурой.

Сформированные информационные предложения (выполняемые SQL-запросы БД) будем понимать как предложения некоторого формального языка. Алфавитом языка являются ИП (контрольные точки), а структура предложений определяется некоторыми правилами.

По результатам синтеза, ИП реализуемого SQL-запроса, можно представить в виде цепочки:

$$s_n = \Pi_{k,1} \oplus \Pi_{k,2} \oplus \dots \oplus \Pi_{k,n}, \quad (1)$$

где n – общее число ИП выполняемого SQL-запроса.

Пусть Z_k – упорядоченное множество ИП некоторого выполняемого SQL-

запроса вычисления (множества выполняемых SQL-запросов) A (рис.1).

$$Z_k = \{ \Pi_{k,1}, \Pi_{k,2}, \dots, \Pi_{k,n} \}, \quad (2)$$

где A – выполняемые SQL-запросы БД;

k – общее число запросов.

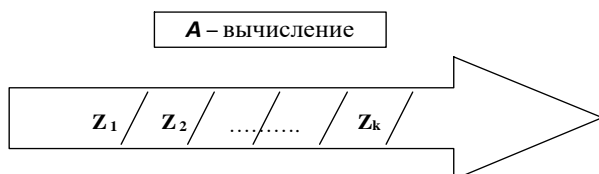


Рис. 1. Множество инвариантов подобия выполняемого SQL-запроса

Грамматика ИП SQL-запросов $G = (N, T, P, S)$, является грамматикой типа 2 или контекстно-свободной грамматикой, если каждое ее правило имеет вид:

$$A \rightarrow \beta \in P,$$

где $A \in N$, $\beta \in (T \cup N)^+$.

На основе введенных понятий построим контекстно-свободную формальную грамматику G в соответствии с отношением (1) для некоторого k -выполняемого SQL-запроса s_k из множества выполняемых SQL-запросов, определив четверку V_N, V_T, P, S следующим образом:

– основной словарь зададим как

$$T := Z_k \cup Q,$$

где Z_k – множество структурных элементов(инвариантов) в соответствии (2);

Q – символ конца вычисления;

– вспомогательный словарь опишем в виде

$$V_N := V_N^* \cup S,$$

где V_N^* – множество нетерминальных символов такое, что существует некоторое взаимно-однозначное соответствие $W_{Z_k V_N^*}$ между Z_k и V_N^* , причем каждый

i -й элемент Z_k отображается в i -й элемент V_N^* , то есть

$$V_N^* = \{ A_i \mid A_i = W_{Z_k V_N^*}(\Pi_{k,i}), i = 1, \dots, n \}.$$

В результате получим:

$$S \rightarrow AB$$

$$A_i \rightarrow a_i A_{i+1} \mid i = 1, \dots, n-1;$$

$$a_i = \Pi_{k,i}, a_i \in V_T, \Pi_{k,i} \in Z_k,$$

$$A_i = F_{Z_k V_N^*}(\Pi_{k,i})$$

$$B_i \rightarrow b_i B_{i+1} \mid i = 1, \dots, n;$$

$$b_i = \Pi'_{k,i}, b_i \in V_T, \Pi'_{k,i} \in Z_k,$$

$$B_i = F_{Z_k V_N^*}(\Pi'_{k,i})$$

$$A_n \rightarrow Q \mid A_n = F_{Z_k V_N^*}(\Pi_{k,n}),$$

где $\Pi_{k,n} \in Z_k$, $i = 1, \dots, n$.

$$B_n \rightarrow b_n \mid B_n = F_{Z_k V_N^*}(\Pi'_{k,n}),$$

где $\Pi'_{k,n} \in Z_k$, $i = 1, \dots, n$.

В таблице представлены данные тестирования демонстрационного прототипа системы обнаружения аномалий, реализованного на основе предложенного метода. Данные приведены в сравнении с существующими системами защиты – такими, как внутренняя подсистема защиты СУБД Oracle (корпорация Oracle) и система «Мираж» (ООО «Институт Компьютерных Технологий», г. Киев).

В условиях реальной работы пользователей с системой обнаружения аномалий, вероятность появления одинакового запроса сразу от нескольких пользователей снижается практически до нуля. Результаты численного моделирования оценки качества методов контроля представлены на рисунке 2.

АНОМАЛИИ (по способам реализаций)	Количество	Предложенный метод	МИРАЖ	ORACLE
UNION, вызывает обращение к другой таблице или представлению	15	15	13	12
Необоснованный SUB-SELECT	15	15	14	14
Зацикливание WHERE	12	12	10	8
Необоснованный вызов хранимых процедур	28	28	21	23
Нелегитимный доступ к системным таблицам и/или пользовательским приложениям	20	18	20	16
Внедрение INSERT, UPDATE и DELETE	10	10	10	8
Пропуск	0	2	12	19
Ложное срабатывание	0	0	15	0

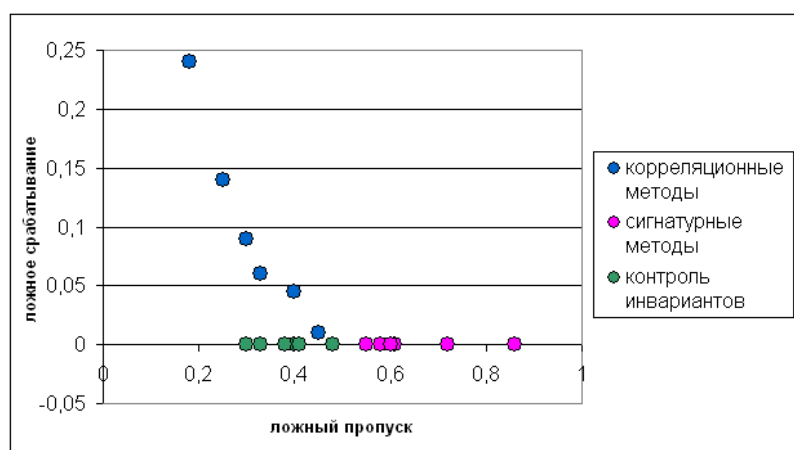


Рис. 2. Результаты численного моделирования оценки качества методов контроля

Заключение

В данной статье в общем виде был рассмотрен метод построения формальной грамматики, представляющей собой модель реализуемого SQL-запроса БД. Которая, в свою очередь, сначала позволяет определить математические свойства ИП выполняемых SQL-запросов, а затем на их базе сформулировать строгие утверждения, необходимые для контроля структурно-функциональной целостности осуществляемых SQL-запросов в условиях внешних дестабилизирующих факторов (различных возмущающих воздействий, сбоев и т.д.).

Описанный метод может быть применен в первую очередь для проверки

БД, предназначенных для сбора, хранения и анализа статистических данных, таких, как контрольные данные о работе программных и аппаратных средств, разных статистических сведений о населении, данных о производстве продукции и т. д. Указанный способ решения задачи эффективен в сравнении с дублированием SQL-запроса и позволяет выполнять обнаружение и коррекцию аномалий SQL-запросов с меньшей трудоемкостью и большей экономичностью.

Предложенный метод использован в решении практической задачи по обнаружению и коррекции аномалий SQL-запросов, позволив эффективнее использовать имеющиеся данные для разностороннего анализа.

Список литературы

1. Дудзенко Д.А., Марковский А.С., Свеколкин Н.И. Обеспечение структурно-функциональной целостности выполняемых SQL-запросов реляционных баз данных на основе инвариантов подобия // Системная инженерия. 2016. № 1-2. С. 56-63.
2. Беляев А.В., Петренко С.А. Безопасность современных корпоративных информационных систем // Защита информации. Конфидент. 2003. № 5. С. 23-30.
3. Петренко С.А., Беляев А.В. Мониторинг безопасности ERP Oracle E-Business Suite. URL: <http://citcity.ru/11190/> (дата обращения 08.02.2017).
4. Маркин А.В. Построение запросов и программирование на SQL: учеб. пособие для вузов по специальности «Информационные системы и технологии». Рязань: РГРТУ, 2008. 312 с.
5. Bertino E., Kamra A., Terzi E., Vakkali A., Intrusion Detection in RBAC-administered Databases. ACSAC '05 Proceedings of the 21st Annual Computer Security Applications Conference, 2005, pp. 170-182.
6. Павлов А.В. Обнаружение аномальной активности в реляционных базах

данных на основе искусственных иммунных систем с отрицательным отбором // Научно-технический вестник Поволжья. 2011. № 1. С. 166-168.

7. Зегжда Д.П., Калинин М.О. Обеспечение доверенности информационной среды на основе расширения понятия «целостность» и управления безопасностью // Проблемы информ. безопасности. Компьютерные системы. 2009. № 4. С. 7-16.

8. Кузнецов С.Д. Базы данных: учебник для студ. учреждений высшего проф. образования. М.: Издательский центр «Академия», 2012. 496 с.

9. Lesov P. Database security: a historical perspective. 2010. URL: <http://arxiv.org/ftp/arxiv/papers/1004/1004.4022.pdf> (дата обращения: 26.05.2017).

10. Смирнов С.Н. Безопасность систем баз данных. М.: Гелиос АРВ, 2007. 352 с.

11. Sandhu Ravi S., Jajodia Sushil. Data and database security and controls. Handbook of Information Security Management, Auerbach Publishers, 1993, pp. 181-199.

Поступила в редакцию 08.01.18

UDC 004.021

A.S. Markovskiy, Candidate of Engineering Sciences, Senior Researcher, Mozhaisky Military Space Academy (Saint-Petersburg, Russia) (e-mail: lexx26@list.ru)

N.I. Svekolkin, Senior Researcher, Mozhaisky Military Space Academy (Saint-Petersburg, Russia) (e-mail: ins_61@mail.ru)

METHOD FOR DETECTING ANOMALIES OF SQL-SPECIFIC QUERY OF RELATIONAL DATABASES

In the context of toughening of the requirements in the field of information security (the conditions of its safety) difficult-to-implement, the increasing number of external destabilizing factors (including the high level of false alarms), the increase of scopes and speed of information changes, and the drawbacks inherent to most databases, the probability of anomalies occurrence in the process of operation (acquisition, processing and storage) of relational databases is high.

The article provides detailed description of the method for the construction a formal grammar executed by a SQL query of relational databases. This approach considers formal grammar under study from a mathematical point of view, as a model that defines a set of discrete objects in the form of description of the original objects and the rules for constructing new objects from the original and already created ones. Thus, a system of rules for further work is

formed, represented in the form of a system of equations. The described method makes it possible to determine mathematical properties of the similarity invariants of the SQL query of relational databases intended for the collection, storage and analysis of statistical data, such as reference data of the operation of software and hardware, various statistical data about population, .production etc.

The results of the testing of the demonstration prototype of the anomaly detection system, implemented on the basis of the proposed method, obtained in the course of the experimental implementation are presented in comparison with some existing and applied security systems. The solution proposed in the article is effective, simple and universal for the majority of currently used relational databases. In addition, it has a low cost of financial expenses in case of practical implementation.

Key words: SQL-specific query, similarity invariant, relational database.

DOI: 10.21869/2223-1560-2018-22-1-53-61

For citation: Markovskiy A.S., Svelkolkin N.I. Method for Detecting Anomalies of SQL-Specific Query of Relational Databases. Proceedings of the Southwest State University, 2018, vol. 22, no. 1(76), pp. 53-61 (in Russ.).

Reference

1. Dudzenko D.A., Markovskiy A.S., Svelkolkin N.I. Obespechenie strukturno-funktsional'noy tselostnosti vypolnyaemykh SQL-zaprosov relyatsionnykh baz dannykh na osnove invariantov podobiya. Sistemnaya inzheneriya, 2016, no. 1-2, pp.56-63.
2. Belyaev A.V., Petrenko S.A. Bezopasnost' sovremennykh korporativnykh informatsionnykh sistem. Zashchita informatsii. Konfident, 2003, no. 5, pp.23-30.
3. Petrenko S.A., Belyaev A.V. Monitoring bezopasnosti ERP Oracle E-Business Suite. URL: <http://citcity.ru/11190/> (08.02.2017).
4. Markin A.V. Postroenie zaprosov i programmirovaniye na SQL: ucheb. posobie dlja vuzov po special'nosti «Informacionnye sistemy i tehnologii». Rjazan': RGRTU Publ., 2008, 312 p.
5. Bertino E., Kamra A., Terzi E., Vakali A., Intrusion Detection in RBAC-administered Databases. ACSAC '05 Proceedings of the 21st Annual Computer Security Applications Conference, 2005, pp. 170-182.
6. Pavlov A.V. Obnaruzhenie anomal'noj aktivnosti v reljacionnykh bazah dannykh na osnove iskusstvennyh immunnyh sistem s otricatel'nym otborom. Scientific and technical Bulletin of the Volga region, 2011, no. 1, pp.166-168.
7. Zegzhda D.P., Kalinin M.O. Obespechenie doverennosti informacionnoj sredy na osnove rasshirenija ponjatija «celostnost'» i upravlenija bezopasnost'ju. Problemy inform. bezopasnosti. Komp'juternye sistemy, 2009, no. 4, pp. 7-16.
8. Kuznetsov S.D. Bazy dannyh: uchebnik dlja stud. uchrezhdenij vysshego prof. obrazovanija. Moscow, Akademiya Publ., 2012, 496 p.
9. Lesov P. Database security: a historicalperspectiv. CoRR, 2010. Available at: <http://arxiv.org/ftp/arxiv/papers/1004/1004.4022.pdf> (26.05.2017).
10. Smirnov S.N. Bezopasnost' sistem baz dannyh. Moscow, Gelios ARV Publ., 2007, 352 p.
11. Sandhu Ravi S., Jajodia Sushil. Data and database security and controls. Handbook of Information Security Management, Auerbach Publishers, 1993, pp. 181-199.