УДК 004.056.2

МЕТОДИКА ОБНАРУЖЕНИЯ И ЛОКАЛИЗАЦИИ СКОМПРОМЕТИРОВАННОГО БЛОКА ДАННЫХ ТАБЛИЦЫ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

Королев И.Д.

д.т.н., профессор, профессор кафедры № 34

Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С.М.Штеменко,

Краснодар, Россия

Литвинов Е.С.

адъюнкт,

Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С.М.Штеменко,

Краснодар, Россия

Мулюкин С.В.

старший оператор,

Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С.М.Штеменко,

Краснодар, Россия

Ахтямов М.О.

старший оператор,

Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С.М.Штеменко,

Краснодар, Россия

Аннотация

Информатизация бизнес-процессов современных предприятий привела к тому, что ущерб от нарушения информационной безопасности может привести к крупным финансовым потерям, вплоть до полного закрытия предприятия. Одной из основных угроз информационной безопасности является отказ в обслуживании баз данных и нарушение целостности хранящейся в них информации. Целью данной работы является разработка способа повышения целостности данных, хранимых в реляционных базах данных. Для этого в работе предлагается модернизировать рассматриваемую таблицу базы данных путём добавления дополнительных строки и столбца, позволяющих отслеживать изменения данных, хранимых в ячейках таблицы. Разработанный способ позволит сэкономить время и вычислительные ресурсы, затрачиваемые на восстановление целостности данных, хранимых в реляционной базе данных.

Ключевые слова: база данных, информационная безопасность, хэш-функция, информационные технологии, защита информации.

A TECHNIQUE FOR DETECTING AND LOCALIZING A COMPROMISED DATA BLOCK OF A RELATIONAL DATABASE TABLE

Korolev I. D.,

Doctor of Technical Sciences, Professor, Professor of the department No. 34

Krasnodar Higher Military Order of Zhukov and the October Revolution Red Banner School named after General of the Army S. M. Shtemenko,

Krasnodar, Russia

Litvinov E. S.

adjunct,

Krasnodar Higher Military Order of Zhukov and the October Revolution Red Banner School named after General of the Army S. M. Shtemenko,

Krasnodar, Russia

Mulyukin S. V.
Senior operator,
Krasnodar Higher Military Order of Zhukov and the October Revolution Red Banner School named after General of the Army S. M. Shtemenko,
Krasnodar, Russia

Akhtyamov M. O.
Senior operator,
Krasnodar Higher Military Order of Zhukov and the October Revolution Red Banner School named after General of the Army S. M. Shtemenko,
Krasnodar, Russia

Annotation
The informatization of business processes of modern enterprises has led to the fact that the damage from the violation of information security can lead to large financial losses, up to the complete closure of the enterprise. One of the main threats to information security is the denial of service of databases and the violation of the integrity of the information stored in them. The purpose of this work is to develop a way to improve the integrity of data stored in relational databases. To do this, we propose to modernize the database table under consideration by adding additional rows and columns that allow you to track changes in the data stored in the table cells. The developed method will save time and computational resources spent on restoring the integrity of data stored in a relational database.

Keywords: database, information security, hash function, information technology, information security.

Стремительное развитие информационных технологий и быстрый рост глобальной сети Интернет привели к формирования информационной среды, оказывающей влияние на все сферы человеческой деятельности. Новые

технологические возможности облегчают распространение информации, повышают эффективность производственных процессов, способствуют расширению деловых операций в процессе бизнеса [1, с. 11].

Применение информационных технологий на современных предприятиях позволяет многократно увеличить эффективность их работы. Однако, стоит отметить, что высокая степень информатизации бизнес-процессов предприятий оказывает не только положительный эффект, но и влечёт за собой новые риски и угрозы. Среди них можно выделить такие угрозы, как:

1. нарушение конфиденциальности;
2. нарушение целостности данных;
3. нарушение доступности [2, с. 742].

Высокий уровень зависимости предприятий от используемых ими электронных ресурсов приводит к неминуемым убыткам в случае реализации описанных выше угроз.

Часто возникающей проблемой в корпоративных информационных системах (далее «ИС») является отказ в обслуживании баз данных и нарушение целостности хранящейся в них информации, вследствие воздействия различных внешних факторов, таких как: воздействие стихийных природных явлений, преднамеренные действия злоумышленников, непреднамеренные действия сотрудников, отказ в работе оборудования и др. [3, с. 16]

Таким образом, проблема восстановления целостности данных, в случае её нарушения, является актуальной по сей день.

Процесс восстановления целостности данных можно декомпозировать на два подпроцесса:

1. локализация скомпрометированного блока данных;
2. восстановление блока данных.

Рассмотрим процесс локализации более подробно. Предположим, что база данных некоторого предприятия представлена взаимосвязанным набором таблиц. При этом присутствует вероятность неконтролируемого изменения значения, хранящегося в некоторой ячейке таблицы. Для восстановления

целостности данных в скомпрометированной ячейке в первую очередь необходимо установить, в какой именно ячейке произошли изменения данных.

На рисунке (рис. 1) представлена структура таблицы БД, состоящая из N полей и M строк.

	1	*2*	...	*N*
1	A_{11}	A_{12}	...	A_{1N}
2	A_{21}	A_{22}	...	A_{2N}
...	...	...	...	...
M	A_{M1}	A_{M2}	...	A_{MN}

Рис. 1 — Структура таблицы БД[1]

Допустим, что значение в ячейке A_{22} было несанкционированно изменено. Необходимо разработать способ, позволяющий зафиксировать факт изменения данных, а также найти измененную ячейку.

Для решения данной задачи предлагается добавить в имеющуюся таблицу дополнительные строку и столбец, содержащие контрольные суммы данных, хранящихся в соответствующих им строках и столбцах таблицы (рис. 2).

Таким образом, в ячейке HV_i будет содержаться значение хэш-функции элементов i-го столбца таблицы, а в HH_j — значение хэш-функции элементов j-ой строки.

В свою очередь, хэш-функция — это функция, применяемая к входящему сообщению произвольной длины и возвращающая выходное значение фиксированной длины. Главной особенностью хэш-функции является однонаправленность, что обуславливает невозможность восстановить исходное сообщение по выходному значению.

[1] Разработано авторами

		1	*2*	…	*N*
		$\boldsymbol{HV_1}$	$\boldsymbol{HV_2}$	**…**	$\boldsymbol{HV_N}$
1	$\boldsymbol{HH_1}$	A_{11}	A_{12}	…	A_{1N}
2	$\boldsymbol{HH_2}$	A_{21}	A_{22}	…	A_{2N}
…	**…**	…	…	…	…
M	$\boldsymbol{HH_M}$	A_{M1}	A_{M2}	…	A_{MN}

Рис. 2 — Структура дополненной таблицы БД[2]

Стоит отметить, что выходное значение хэш-функции для одного и того же набора входных данных всегда будет одинаковым. Это позволяет фиксировать факт изменения входных данных (значения строки / столбца таблицы) путём сравнения полученного значения хэш-функции с эталонным [4, с. 37].

Функция хэширования должна обладать следующими свойствами:

1. Хэш-функция может быть применена к аргументу любого размера.
2. Выходное значение хэш-функции имеет фиксированный размер.
3. Хэш-функция должна быть чувствительная к всевозможным изменениям во входном тексте, таким как вставки, выбросы, перестановки и т.п.
4. Хэш-функция должна быть однонаправленной, то есть обладать свойством необратимости.
5. Вероятность того, что значения хэш-функций двух различных документов совпадут, должна быть ничтожно мала [1, с. 148].

Благодаря введению в таблицу ячеек (HH_j), содержащих значения хэш-функций строк и ячеек (HV_i), содержащих значения хэш-функций столбцов, вместе с изменением ячейки A_{22} изменятся и значения соответствующих ячеек (HH_2 и HV_2), как показано на рисунке (рис. 3). Таким образом, появится возможность локализовать скомпрометированный блок данных, путём анализа изменения ячеек, содержащих значения хэш-функций строк и столбцов.

[2] Разработано авторами

		1	*2*	...	*N*
		HV₁	***HV₂***	**...**	***HV$_N$***
1	***HH₁***	A_{11}	A_{12}	...	A_{1N}
2	***HH₂***	A_{21}	**A_{22}**	...	A_{2N}
...	**...**	...	...	...	...
M	***HH$_M$***	A_{M1}	A_{M2}	...	A_{MN}

Рис. 3 — Структура дополненной таблицы БД[3]

Принимая во внимание условие, что изменено может быть значение исключительно одной ячейки таблицы, можно быть уверенными в том, что изменятся только два значения хэш-функций — хэш-функция столбца и хэш-функция строки, к которым относится изменённая ячейка.

Рассмотрим практический пример применения вышеописанного способа.

Имеется исходная таблица, хранящая данные сотрудников некоторого отдела (рис. 4). Как можно заметить, таблица уже дополнена ячейками HV и HH, хранящих значения хэш-функций соответствующих им строк и столбцов.

		ID	*Имя*	*Фамилия*	*Возраст*
		HV₁	***HV₂***	***HV₃***	***HV₄***
1	***HH₁***	1	Иван	Иванов	24
2	***HH₂***	2	Павел	Павлов	32
3	***HH₃***	3	Максим	Максимов	21
4	***HH₄***	4	Пётр	Петров	24

Рис. 4 — Исходная таблица с данными [4]

Рассчитаем значения ячеек HV и HH для таблицы с эталонными данными (рис. 4). Для расчёта будем использовать хэш-функцию md5. Решение об использовании данного алгоритма было принято в связи с тем, что его выходное значение имеет размер 128 бит, в то время, как самый легковесный алгоритм семейства SHA - SHA-1 имеет размер выходного значения равный 160

[3] Разработано авторами
[4] Разработано авторами

бит. Это позволит минимизировать объём памяти, затрачиваемый на хранение ячеек со значениями хэш-функций.

В таблице (таблица 1) показаны значения, полученные в результате расчёта хэш-функций.

Таблица 1 — Значения ячеек HV и HH

Имя ячейки	Значение хэш-функции
HV_1	0d5cdb236e1063a40e245f43edd4e8c0
HV_2	10d85ea4980aefd1fcaaa53957254893
HV_3	e33b9bc82771f84c7f48aeb3181edd30
HV_4	2a8b82fe2c56e4efe284f2446c575867
HH_1	aedd39b5a5100f1c625af4bd80bffa9f
HH_2	0774b6b4858fa6692ee0a8fdbc530378
HH_3	abb39eb7711ae5e2943feab60f707975
HH_4	8d4be2f01212a04d8f4f28cb8d2d835a

Допустим, что в ходе компьютерной атаки на информационную систему было несанкционированно изменено значение поля «Имя» в строке с номером «3». Таким образом исходная таблица приняла вид, показанный на рисунке (рис. 5).

Необходимо определить, в какой именно ячейке таблицы произошло изменение и восстановить исходное значение. Для этого произведём перерасчёт значений ячеек, хранящих хэши, для изменённой таблицы, в ходе которого увидим, что все значения остались неизменными, за исключением ячеек HV_2 и HH_3 (таблица 2).

		ID	*Имя*	*Фамилия*	*Возраст*
		HV_1	***HV_2***	***HV_3***	***HV_4***
1	***HH_1***	1	Иван	Иванов	24
2	***HH_2***	2	Павел	Павлов	32
3	***HH_3***	3	Степан	Максимов	21
4	***HH_4***	4	Пётр	Петров	24

Рис. 5 — Скомпрометированная таблица[5]

[5] Разработано авторами

Исходя из полученной информации можно сделать вывод, что изменённая ячейка таблицы принадлежит к столбцу и строке, значения хэш-функций которых хранятся в ячейках HV_2 и HH_3.

Таблица 2 — Значения изменившихся ячеек

Имя ячейки	Значение хэш-функции
HV_2*	fc15703af0459238d55844fc71ef7274
HH_3*	179a881779f0307c908c400bc0d90b75

Зная, в какой именно ячейке произошли изменения, мы можем точечно восстановить исходное значение (например путём его загрузки из удалённого хранилища эталонных данных), что позволит сэкономить время и вычислительные ресурсы, затрачиваемые на восстановление целостности данных, хранимых в реляционной базе данных.

Библиографический список:

1. Шаньгин В.Ф., Защита компьютерной информации. Эффективные методы и средства / Шаньгин В.Ф. - М.: ДМК Пресс, 2010. – С. 544.
2. Олифер В. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд / Олифер В., Олифер Н. — СПб.: Питер, 2017. – С. 745.
3. Шаньгин В.Ф., Информационная безопасность компьютерных систем и сетей: учеб. Пособие / Шаньгин В.Ф. - М.: ИД «ФОРУМ»: ИНФРА-М, 2008. - С. 544.
4. Шнайер Б., Прикладная криптография. Протоколы, алгоритмы и исходные тексты на языке С: 2-е изд. / Шнайер Брюс. - С. 610.

Оригинальность 78%