УДК 004.056.53

БЕЗОПАСНОСТЬ ЛОКАЛЬНЫХ БАЗ ДАННЫХ НА ПРИМЕРЕ SQL SERVER COMPACT

© А.Е. Потапов, Д.В. Манухина, И.А. Соломатина, А.С. Нилова, А.И. Бадмаев, А.В. Яковлев

Ключевые слова: безопасность; базы данных; SQL Server Compact.

Рассмотрена проблема защиты локальных баз данных от несанкционированного доступа. Дано сравнение методов, позволяющих избежать хранения конфиденциальной информации в виде, допускающем несанкционированный доступ к данным. Особое внимание уделено необходимости шифрования разделов конфигурационных файлов приложений, содержащих информацию для подключения к источнику данных. На основе проведенного исследования сделан вывод, что уровень защиты локальной базы данных зависит от требований по безопасности к разрабатываемому приложению.

В настоящее время базы данных являются ключевыми компонентами большинства клиент-серверных приложений. Причем, кроме центральной базы данных на сервере, зачастую применяется клиентская локальная база данных. Такой подход обеспечивает полную или частичную автономность приложения, а также увеличивает скорость его работы, т. к. большинство данных, необходимых для работы, находится на стороне клиентского приложения, и, таким образом, пропадает необходимость в обращениях к удаленной базе данных. Однако на стороне клиентского приложения может храниться конфиденциальная информация, поэтому обеспечение безопасности локальных баз данных является одной из важнейших задач при проектировании такого типа приложений. Под безопасностью баз данных подразумевается защита находящейся в ней конфиденциальной информации. Для обеспечения безопасности базы данных необходимо защитить ее от нелегального доступа, а также от любой вредоносной или случайной модификации данных [1].

Самый простой способ защитить базу данных – это установить пароль для подключения к базе. В случае с SQL Server Compact (реляционная база данных компании Microsoft, хранящаяся в одном файле и распространяемая бесплатно) после установки пароля автоматически включается шифрование файла базы данных, т. к. использование шифрования без пароля или пароля без шифрования не имеет смысла.

Однако по умолчанию строка подключения, содержащая всю необходимую информацию для подключения к базе данных, хранится в конфигурационном файле, который находится, как правило, в папке с исполняемым файлом приложения и может быть открыт и прочитан любым пользователем. Для приложений, к которым предъявляются требования по безопасности, такая ситуация недопустима. Существует несколько способов, позволяющих избежать хранения конфиденциальных данных в открытом виде.

1. Хранение или генерация пароля в коде приложения. При этом пароль не хранится в конфигурационном файле, а подставляется напрямую из кода путем изменения строки подключения к базе данных. Способ изменения строки подключения зависит от выбранного провайдера для SQL Server Compact.

Однако данный подход к решению проблемы безопасности имеет некоторые недостатки. Во-первых, пароль для всех копий клиентского приложения будет одинаковым. А во-вторых, исходный код приложения может быть просмотрен с помощью специализированной программы – дизассемблера. Таким образом, открыв исполняемый файл или библиотеку, которая содержит пароль к базе данных, с помощью дизассемблера злоумышленник может легко подключиться к локальной базе данных.

2. Получение пароля для подключения к локальной базе данных от сервера. Данный метод защиты похож на предыдущий, однако пароль (или метод его генерации) не хранится в коде приложения в открытом виде. К тому же для каждой клиентской базы данных может генерироваться уникальный пароль, зависящий, например, от параметров рабочего места клиента. При реализации данного способа желательно использовать криптозащищенный канал связи клиента с сервером, т. к. в противном случае пароль может быть перехвачен и прочитан при передаче.

К недостаткам данного подхода можно отнести то, что вход в приложение будет возможен только при наличии связи с сервером. Это накладывает некоторые ограничения на автономность приложения, однако не мешает работе при нестабильном соединении.

3. Шифрование секции со строкой подключения в конфигурационном файле. Метод защиты конфигурации можно использовать для шифрования конфиденциальной информации, включая имена и пароли пользователей, строки подключения к базам данных и ключи шифрования, в файле конфигурации. Конфигурационный файл, в котором значения строк подключения зашифрованы с помощью метода защиты конфигурации, не показывает строки подключения открытым текстом, а хранит их в зашифрованном виде.

Шифрование и расшифровка содержимого файла конфигурации выполняется с помощью классов, насле-

дующихся от *ProtectedConfigurationProvider*. В .Net Framework включены два поставщика защищенной конфигурации: *DpapiProtectedConfigurationProvider* и *RsaProtectedConfigurationProvider*. Оба поставщика обеспечивают надежное шифрование данных, однако если планируется использовать один зашифрованный файл конфигурации на нескольких компьютерах, то только *RsaProtectedConfigurationProvider* позволяет экспортировать ключи шифрования и импортировать их на другом компьютере.

Класс *DpapiProtectedConfigurationProvider* использует *DPAPI* (Data Protection application programming interface) – криптографический интерфейс программирования приложений в ОС семейства Windows, обеспечивающий защиту (конфиденциальность) данных путем их шифрования. Данный поставщик может быть сконфигурирован для защиты, ориентированной на машину или учетную запись пользователя.

Класс *RsaProtectedConfigurationProvider* использует функции шифрования, предоставленные классом *RSA* (аббревиатура от фамилий Rivest, Shamir и Adleman), который, в свою очередь, обеспечивает реализацию ключа *RSA*. *RSA* – криптографический алгоритм, надежность которого основывается на трудности факторизации больших чисел и вычислении дискретных логарифмов. Данный алгоритм шифрования относится к классу асимметричных и использует два простых случайных числа для генерации открытого и закрытого ключей [2].

Платформа .Net Framework позволяет использовать собственного поставщика защищенной конфигурации. Целесообразно реализовывать собственного поставщика при необходимости использовать алгоритм шифрования, отличный от предоставляемых классами *DpapiProtectedConfigurationProvider* и *RsaProtectedConfigurationProvider*.

4. Запрашивать пароль к базе данных при входе клиента в приложение. Данный способ позволяет избежать хранения пароля в файле конфигурации, но предоставляет пользователю возможность подключаться к базе данных не только посредством клиентского приложения, но и с помощью сторонних программ. В этом случае пользователь может напрямую подключиться к локальной базе данных и получить доступ к конфиденциальной информации, скрытой от него средствами клиентского приложения.

Приведенные способы обеспечивают разные уровни защиты локальных баз данных. Выбирать метод следует исходя из требований по безопасности к приложению, т. к. с повышением уровня защиты увеличиваются и затраты на реализацию алгоритма.

ЛИТЕРАТУРА

1. *Basharat I., Azam F., Muzaffar A.W.* Database Security and Encryption: A Survey Study // International Journal of Computer Applications. 2012. V. 47. № 12. P. 28-34.
2. *Singh G., Supriya* A study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security // International Journal of Computer Applications. 2013. V. 67. № 19. P. 33-38.

Поступила в редакцию 17 апреля 2014 г.

Potapov A.E., Manuhina D.V., Solomatina I.A., Nilova A.S., Badmayev A.I., Yakovlev A.V. SECURITY OF LOCAL DATABASES ON EXAMPLE OF SQL SERVER COMPACT

This article discusses the problem of protecting local databases from unauthorized access. A comparison of methods that allow avoiding storing confidential information to prevent any unauthorized access to data is provided. Particular attention is paid to the necessity encrypt sections of the configuration file applications that contain information for connecting to the data source. Based on of the research it is concluded that the level of protection of the local database depends on the safety requirements to develop applications.

Key words: security; database; SQL Server Compact.

Потапов Андрей Евгеньевич, Московский государственный технический университет им. Н.Э. Баумана, Калужский филиал, г. Калуга, Российская Федерация, кандидат физико-математических наук, доцент кафедры систем автоматизированного проектирования, e-mail: jerry1st@mail.ru

Potapov Andrey Evgenyevich, Bauman Moscow State Technical University, Kaluga branch, Kaluga, Russian Federation, Candidate of Physics and Mathematics, Associate Professor of Systems of Automated Simulation Department, e-mail: jerry1st@mail.ru

Манухина Дарья Владимировна, Московский государственный технический университет им. Н.Э. Баумана, Калужский филиал, г. Калуга, Российская Федерация, кандидат физико-математических наук, доцент кафедры систем автоматизированного проектирования, e-mail: dmanuhina@gmail.com

Manuhina Darya Vladimirovna, Bauman Moscow State Technical University, Kaluga branch, Kaluga, Russian Federation, Candidate of Physics and Mathematics, Associate Professor of Systems of Automated Simulation Department, e-mail: dmanuhina@gmail.com

Соломатина Ирина Алексеевна, Московский государственный технический университет им. Н.Э. Баумана, Калужский филиал, г. Калуга, Российская Федерация, студентка факультета электроники, информатики и управления, кафедра систем автоматизированного проектирования, e-mail: solomatina.irinka@yandex.ru

Solomatina Irina Aleksyevna, Bauman Moscow State Technical University, Kaluga branch, Kaluga, Russian Federation, Student of Electronics, Informatics and Management Faculty, Systems of Automated Simulation Department, e-mail: dmanuhina@gmail.com

Нилова Анастасия Сергеевна, Московский государственный технический университет им. Н.Э. Баумана, Калужский филиал, г. Калуга, Российская Федерация, студентка факультета электроники, информатики и управления, кафедра систем автоматизированного проектирования, e-mail: dyudyuka.barbidonskaya@yandex.ru

Nilova Anastasia Sergeyevna, Bauman Moscow State Technical University, Kaluga branch, Kaluga, Russian Federation, Student of Electronics, Informatics and Management Faculty, Systems of Automated Simulation Department, e-mail: dyudyuka.barbidonskaya@yandex.ru

Бадмаев Алексей Игоревич, Московский государственный технический университет им. Н.Э. Баумана, Калужский филиал, г. Калуга, Российская Федерация, студент факультета электроники, информатики и управления, кафедра систем автоматизированного проектирования, e-mail: alexey_bdmv@mail.ru

Badmayev Aleksey Igorevich, Bauman Moscow State Technical University, Kaluga branch, Kaluga, Russian Federation, Student of Electronics, Informatics and Management Faculty, Systems of Automated Simulation Department, e-mail: alexey_bdmv@mail.ru

Яковлев Алексей Владимирович, Тамбовский государственный университет им. Г.Р. Державина, г. Тамбов, Российская Федерация, кандидат физико-математических наук, доцент кафедры общей физики, e-mail: feodorov@tsu.tmb.ru

Yakovlev Aleksey Vladimirovich, Tambov State University named after G.R. Derzhavin, Tambov, Russian Federation, Candidate of Physics and Mathematics, Associate Professor of General Physics Department, e-mail: feodorov@tsu.tmb.ru