DOI: 10.17117/na.2016.01.01.346

Поступила (Received): 24.01.2016

<http://ucom.ru/doc/na.2016.01.01.346.pdf>

**Амантай А.Ж., Боярский Э.Ф., Амиров А.Ж.
Защищенность локальных баз данных на примере
использования SQL SERVER COMPACT**

**Amantaj A.Zh., Boyarskij E.F., Amirov A.Zh.
Security of local databases by the example of
the use of SQL SERVER COMPACT**

В статье рассмотрена проблема защиты локальных баз данных от НСД. 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

Ключевые слова: безопасность, базы данных, SQL server compact, security, database

Амантай Ардак Жанатулы

Магистрант

Карагандинский государственный технический университет

г. Караганда, Бульвар-мира, 56

Боярский Эдуард Федорович

Кандидат технических наук, доцент

Карагандинский государственный технический университет

г. Караганда, Бульвар-мира, 56

Амиров Азамат Жанбулатович

Заведующий кафедрой

Карагандинский государственный технический университет

г. Караганда, Бульвар-мира, 56

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, security, database

Amantaj Ardak Zhanatuly

Master

Karaganda state technical university

Karaganda, Boulevard Mira, 56

Bojarskij Eduard Fedorovich

Candidate of Technical Sciences, Associate Professor

Karaganda state technical university

Karaganda, Boulevard Mira, 56

Amirov Azamat Zhanbulatovich

Head of Department

Karaganda state technical university

Karaganda, Boulevard Mira, 56

Сегодня базы данных считаются основными составляющими большинства клиент-серверных приложений. Кроме того, помимо центральной

базы данных на сервере, часто используется клиентская локальная база данных. Такой подход гарантирует полную или частичную автономность приложения, также повышает скорость его работы, т. к. большая часть данных, нужных для работы, располагается на стороне клиентского приложения, и, следовательно, пропадает надобность в обращениях к удаленной информационной базе. Впрочем на стороне клиентского приложения может храниться конфиденциальная информация, потому обеспечение защищенности локальных баз данных считается одной из главных задач при конструировании такого вида приложений. Под защищенностью баз данных имеется в виду защита находящейся в ней конфиденциальной информации. Для обеспечения защищенности информационной базы необходимо защитить ее от нелегального доступа, а также от любой вредоносной или случайной модификации данных [1].

Самый обычный метод защитить информационную базу – 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) в последствии установки пароля автоматически срабатывает шифрование файла информационной базы, т. к. применение шифрования в отсутствии пароля или пароля без шифрования не имеет толку.

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 методом изменения строки подключения к информационной базе. Метод изменения строки подключения находится в зависимости 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 информационной базе от сервера. Данный способ защиты похож на предшествующий, впрочем пароль (или способ его генерации) не хранится в коде приложения в открытом виде. К тому же для каждой клиентской информационной базы может генерироваться оригинальный пароль, зависящий, к примеру, от характеристик рабочего места клиента. При реализации данного метода предпочтительно применять криптозащищенный канал связи клиента с сервером, т. к. иначе 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. Способ защиты конфигурации применяют 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 при помощи способа защиты конфигурации, не демонстрирует строки подключения открытым текстом, а хранит их в зашифрованном виде.

Шифрование и расшифровка содержимого файла конфигурации выполняется с помощью классов, наследующихся 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 дает возможность использовать собственного поставщика защищенной конфигурации. Имеет смысл реализовывать собственного поставщика при необходимости применять метод шифрования, отличный от предоставляемых классами `DpapiProtectedConfigurationProvider` и `RsaProtectedConfigurationProvider`.

4. Запрашивать пароль к информационной базе при входе клиента в приложение. Этот метод позволяет недопустить хранения пароля в файле конфигурации, хотя дает пользователю возможность подключаться к информационной базе не только при помощи клиентского приложения, но и при помощи сторонних программ. В такой ситуации пользователь имеет возможность напрямую подключиться к локальной информационной базе 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.

Список используемых источников:

1. Basharat I., Azam F., Muzafar A.W. Database Security and Encryption // *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.
3. Вербицкий С.Н., Рыков В.В. Численное исследование оптимальных политик управления скоростью обслуживания // *Автоматика и телемеханика*. 1998. № 11. С. 59-70.
4. Ажмухамедов И.М. Динамическая нечеткая когнитивная модель влияния угроз на информационную безопасность системы // *Безопасность информационных технологий*. 2010. №2. С. 68-72.

© 2016, Амантай А.Ж., Боярский Э.Ф., Амиров А.Ж.
Защищенность локальных баз данных на примере
использования SQL SERVER COMPACT

© 2016, Amantaj A.Zh., Boyarskij E.F., Amirov A.Zh.
Security of local databases by the example of the use
of SQL SERVER COMPACT