Настройка прав доступа на базу данных

Фленов Михаил http://www.vr-online.ru

Безопасность сервера во многом зависит от того, как хорошо вы сможете настроить права доступа на объекты. Если где-то предоставить пользователю чуть-чуть лишнего, так сразу жди проблем. Нет, пользователь не будет использовать твои ошибки. Ими воспользуюсь я, или другой хакер. И тогда распрощайся со своими таблицами с данными или всей базой данных. Наша жизнь беспощадна не только в реале, но и в виртуале.

Введение

Почему-то под безопасностью базы данных подразумевается защита от вторжения извне, т.е. совершенное злостным хакером, напившимся бочкой пива :). Нет, такие взломы происходят слишком редко. Я работаю сейчас программистом в достаточно крупной конторе, и администратор вообще не задумывается о защите портов сервака, на котором открыто все, что угодно. На одном сервере крутится куча баз, программ и даже FTP сервер и за 5 лет его ни разу не взломали :). Благо я уломал этого админа установить WEB сервер на отдельное железо, а то если бы народ узнал IP адрес нашего главного сервера, то его любой ламер смог бы проскриптить. Ни база данных, ни Windows не патчились уже несколько лет.

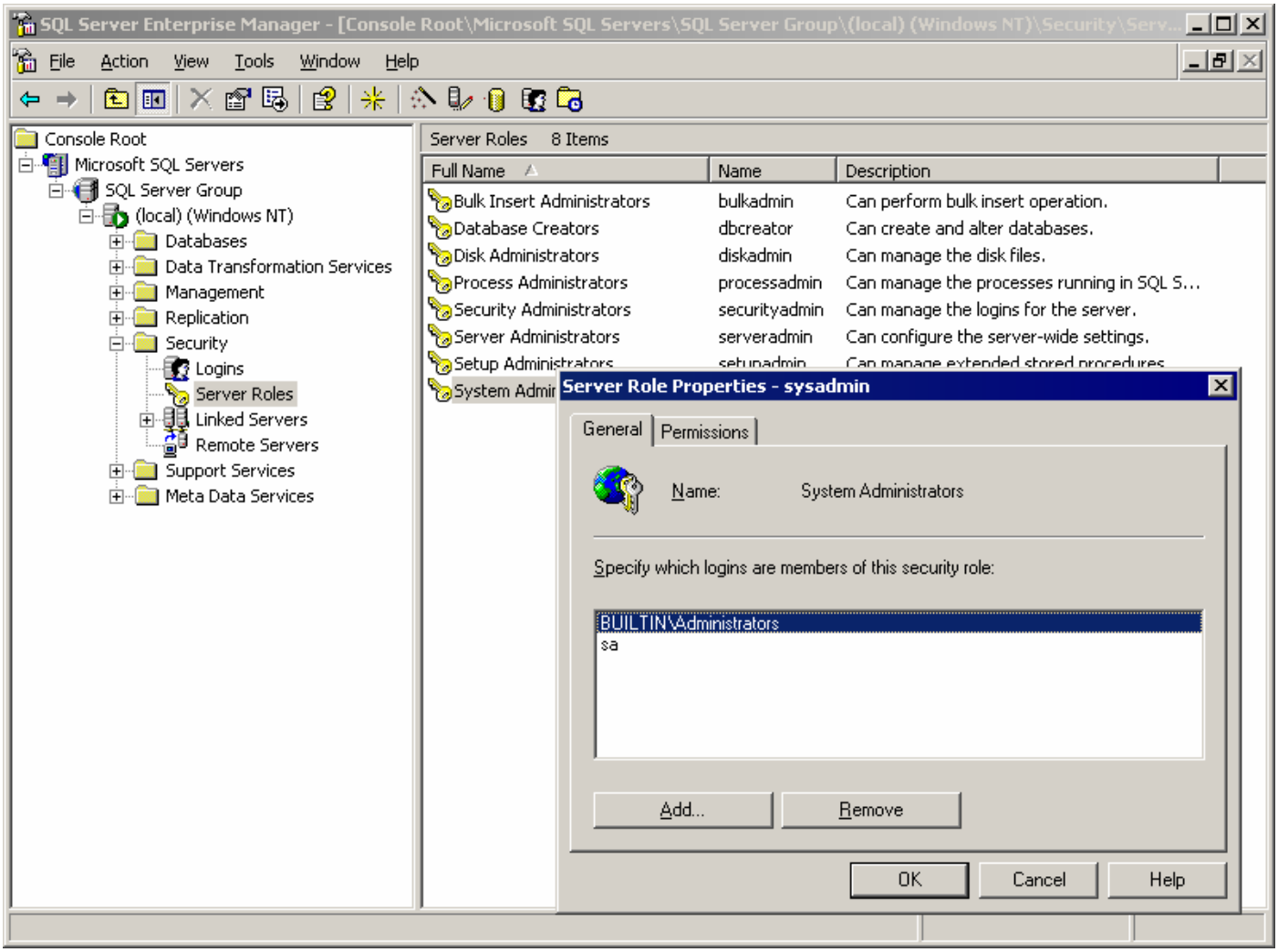
А вот внутренние проблемы из-за неправильной политики безопасности возникают каждый день. Все пользователи входят в систему с правами администратора и могут творить, что угодно. Это действительно проблема, потому что излишние права дают ламерам возможность показать свою безграмотность в полной мере. Поэтому мы будем рассматривать безопасность вне зависимости от того, откуда идет угроза – извне (от хакера) или изнутри (от ушастого юзера).

Чтобы у тебя не было проблем, мы рассмотрим все необходимые основы безопасности не только защиты от хакеров, но и от особо ушастых. В качестве примера я выбрал MS SQL Server, потому что он содержит все, что есть в других базах (Oracle, MySQL и т.д.) и имеет дополнительные возможности по управлению безопасностью. Кто-то может тут подумать, что это делает MS круче. Нет, дополнительные возможности иногда избыточны и только добавляют проблем.

Серверные роли

В Windows и других ОС для управления правами существуют группы и пользователи. С помощью групп мы можем объединять пользователей в кучу и назначать права им всем сразу. Это проще, чем назначать права каждому в отдельности. В базах данных для этих целей существует понятие «роль». Допустим, что 100 пользователей должны иметь право читать данные из определенной таблицы. Давать каждому из них это право напряжесто. Намного проще создать роль, которой разрешено читать, а потом всех нужных пользователей включить в нее. Результат подобен группировке.

В SQL сервере бывает два типа ролей – серверные и баз данных (вторые мы рассмотрим позже). Серверные роли заранее определены, и их изменять нельзя. Открой в Enterprise Manager ветку Security/Server role, и в правой части окна можно будет увидеть список встроенных ролей. Что может делать пользователь соответствующей роли можно определить по описанию.



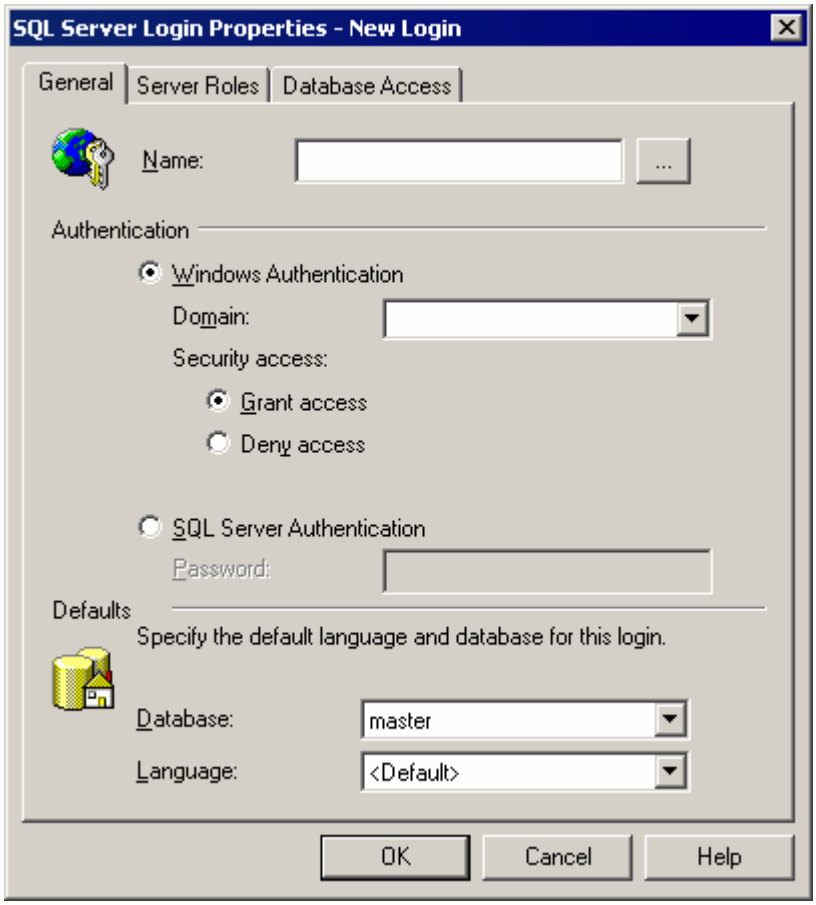
Серверные роли в MS SQL Server и окно редактирования роли

Для добавления уже существующего пользователя в роль, нужно щелкнуть по строке роли дважды и в появившемся окне можно добавлять пользователей в роль или удалять. На закладке Permission более подробно описано, что может делать выделенный пользователь.

Пользователи

Для управления пользователями открой в Enterprise Manager ветку Security/Logins. В правой части появится список всех пользователей сервера. По умолчанию доступ имеют администраторы домена и встроенная учетная запись sa.

Для добавления нового пользователя, щелкни правой кнопкой в пустом месте правой половины окна и в появившемся меню выбери New login. Перед нами откроется окно добавления нового пользователя. В самом верху окна выбирается имя пользователя. Если нужно выбрать уже существующего юзера домена или компьютера, то щелкни по кнопке (…) справа от поля ввода, и ты увидишь окно поиска пользователя в домене.



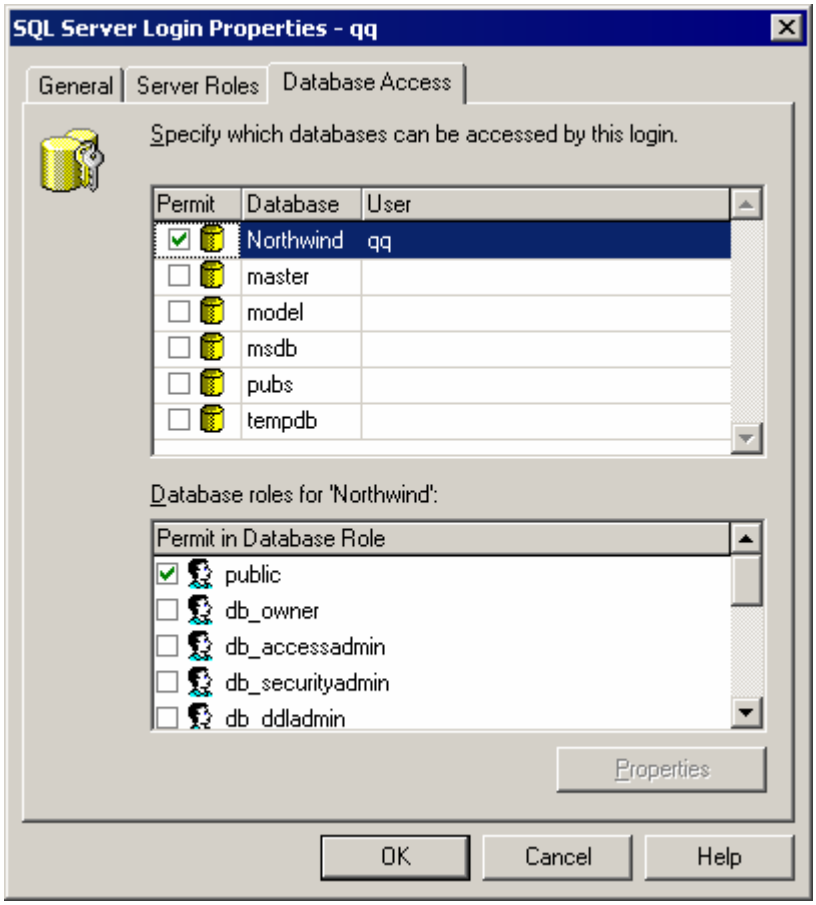
Добавление нового пользователя

Чуть ниже выбирается тип аутентификации – Windows или SQL Server. Если выбрать Windows, то пароль указывать не надо, потому что сервер сам возьмет его из системы. Но зато можно выбрать один из переключателей – Grant access (разрешить доступ) или Deny access (запретить). Во втором случае пользователь будет прописан в базе, но подключиться он не сможет – запрещено однако.

Если выбрать аутентификацию SQL Server, то нужно будет задать пароль, потому что в этом случае, он будет храниться в системных таблицах сервера баз данных. Обрати внимание – даже если в настройках сервера указана только аутентификация Windows, записи SQL сервера создавать разрешено, но вот войти в систему с этими записями будет невозможно.

При определении прав доступа, первым делом нужно помнить закон – что не разрешено, то запрещено. Лучше лишний раз запретить, чем оставить без внимания.

На закладке Server Roles можно указать, какой серверной роли будет принадлежать юзер. Таким образом, уже на этапе создания можно включить юзеров в нужные роли.



Доступ пользователя к базам данных

На закладке Database Access указываем базы, с которыми может работать юзер. Здесь окно разделено на две части: в верхней половине можно выбирать базу данных, к которой разрешен доступ, а в нижнем списке выбирается роль базы данных. В зависимости от выбранной роли в базе, пользователю будут доступны те или иные права. Один пользователь может входить в несколько ролей.

Давай для примера создадим учетную запись qq, которой будет разрешен доступ к базе данных Northwind. Это стандартная тестовая база данных, которая создается при установке сервера. Сохрани изменения. Теперь открой ветку Databases/Northwind/Users и увидишь список пользователей, которым разрешен доступ к выбранной базе данных. Обрати внимание, что запись qq присутствует здесь. В других базах она будет отсутствовать, потому что к ним доступ нашего нового пользователя запрещен.

> У спеца по безопасности не должно быть друзей. Если ты дашь другу высокие права, то он по случайности или ради шутки может подвести доверие. А оно нам надо?

Роли баз данных

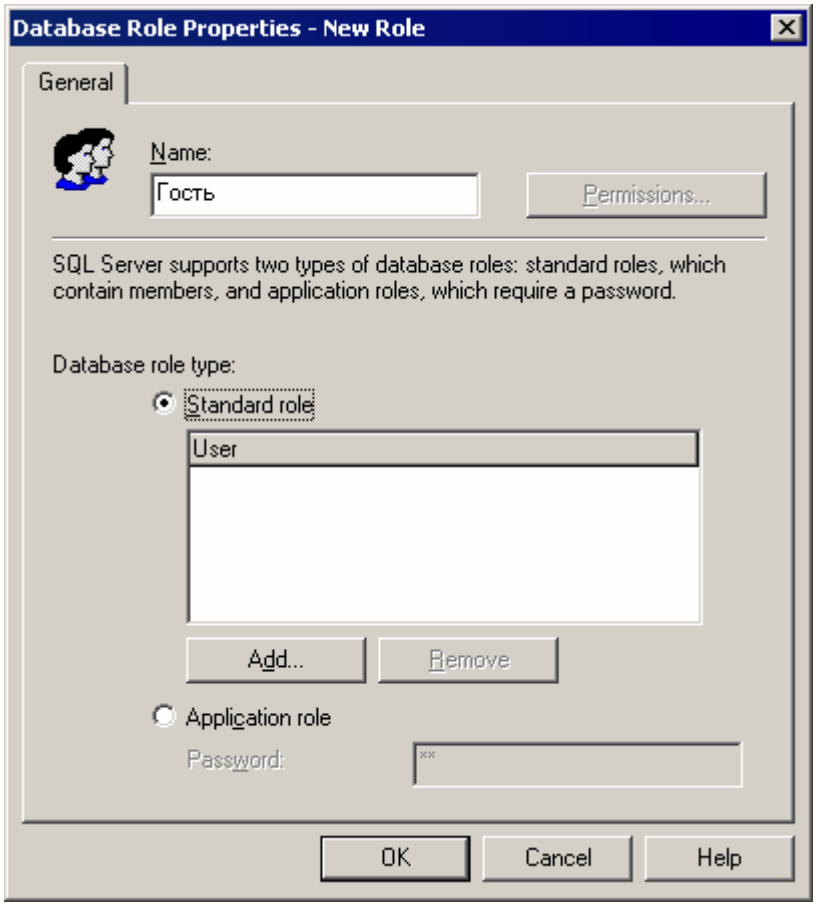
У каждой базы данных могут быть свои роли, которые определяют права доступа на объекты. Многие администраторы не любят возиться с этими правами, а всем устанавливают встроенный по умолчанию public, который позволяет практически все. Если прав роли public не хватает, то тогда просто включают пользователя в серверную роль System Administrator. И вот тут база данных становиться уязвимой по полной программе.

Каждому пользователю должны даваться свои права, в которых разрешены только необходимые действия, а то, что не разрешено, то обязательно должно быть запрещено. Роли, которые уже существуют в сервере использовать нельзя, потому что

их права слишком демократичны. Чтобы они не смущали, лучше даже удалить их все, особенно всеми любимый public.

Создание роли

Для создания новой роли базы данных щелкни правой кнопкой по ветке Databases/Имя базы/Roles и в появившемся меню выбери пункт New database role. Перед нами открывается окно создания новой роли. В самом верху окна нужно ввести имя роли. Например, мы хотим создать роль для бухгалтеров фирмы. Для этого введем имя Buh.

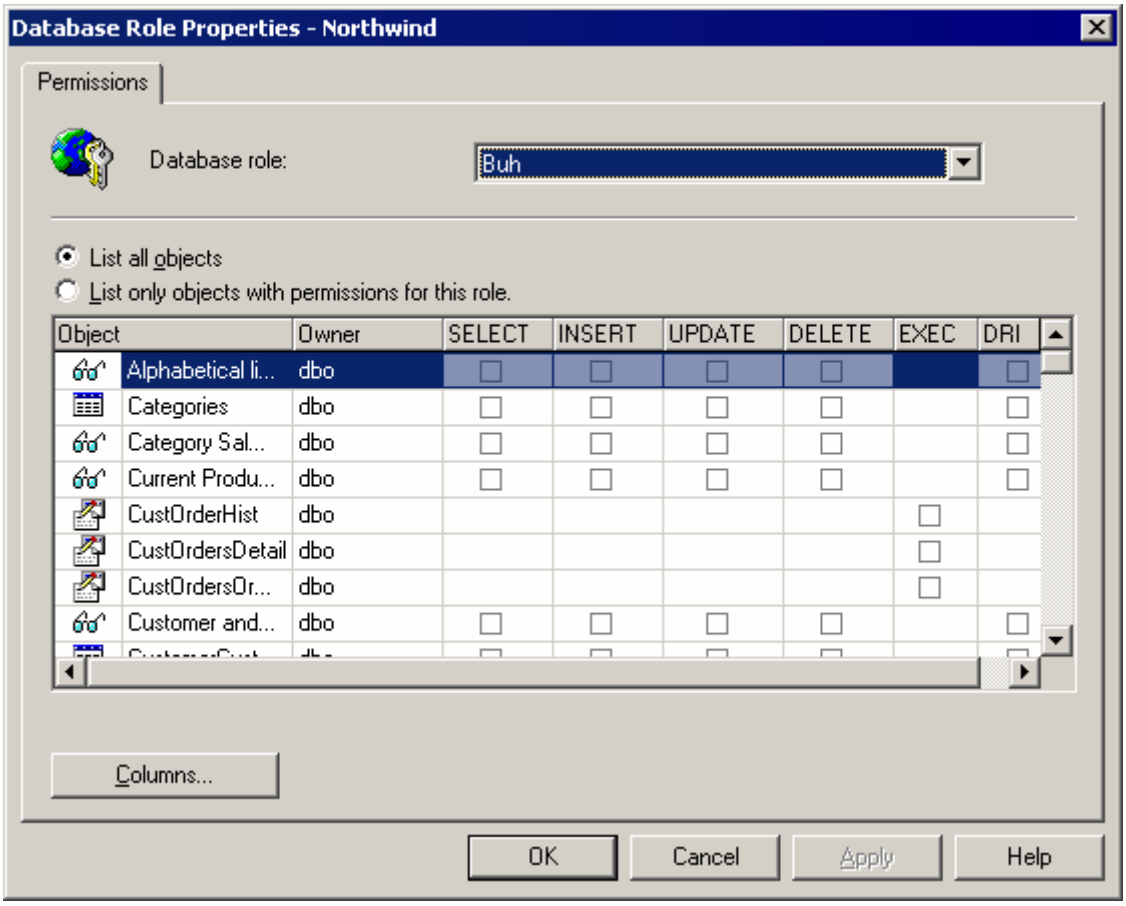


Создание роли базы данных

Чуть ниже выбирается тип роли. Мы остановимся на стандартной, которая выбрана по умолчанию. При этом в центре окна есть список пользователей, которые будут входить в роль. Сейчас список пуст, но если нажать кнопку Add, то можно добавить пользователей. Добавим для примера созданного ранее qq. Больше ничего сделать на этапе создания роли нельзя. Сохраняй изменения нажатием OK.

Права доступа

Теперь посмотрим, как можно назначать права доступа. Дважды щелкни по созданной ранее роли buh и снова откроется окно, которое было при создании, но в этот раз оно открылось для редактирования. Обрати внимание, что кнопка Permission стала доступной, чего не было раньше. Только когда роль уже прописана в базе, можно изменять ее права. Щелкаем по этой кнопке и видим окно настройка прав на объекты базы данных.



Настройка прав доступа в ролях

Вверху окна находится список ролей базы, чтобы можно было быстро переключаться между ними, а сейчас там выбрана роль Buh. В центре окна находится большая сетка из следующих колонок:

- Object имена объектов;

- Owner владелец объекта;

- SELECT разрешение на просмотр данных или выполнение команды SELECT. Доступно только для таблиц и вьюшек;

- INSERT разрешение на добавление данных или выполнение команды INSERT. Доступно только для таблиц и вьюшек;

- UPDATE разрешение на изменение данных или выполнение команды UPDATE. Доступно только для таблиц и вьюшек;

- DELETE разрешение на удаление данных или выполнение команды DELETE. Доступно только для таблиц и вьюшек;

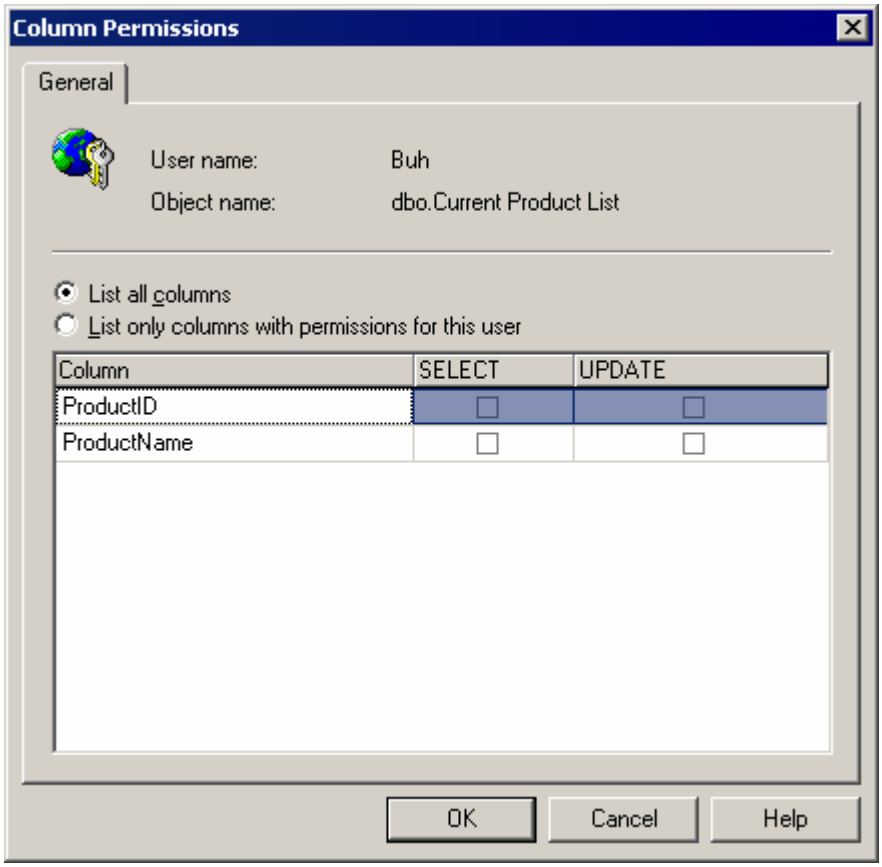
- EXEC разрешение на выполнение хранимых процедур и функций. Доступно только для хранимых процедур и функций;

- DRI (declarative referential integrity) обеспечение целостности. Доступно только для таблиц, вьюшек и функций.

В новой роли никаких прав нет. Чтобы добавить возможность просмотра таблицы, например, Categories, нужно щелкнуть в квадрате на пересечении строки и Categories и колонки SELECT. Щелчок устанавливает в этом квадрате зеленую галочку, что соответствует разрешению. Второй щелчок меняет галочку на красный крест, что соответствует запрету. Это бывает необходимо, если пользователь может получить

доступ, если он находиться одновременно и в другой роли, где доступ на выбранное действие разрешен. Третий щелчок снимает какие-либо разрешения на действие и оставляет квадрат пустым. Это означает, что доступа нет, но он может быть делегирован, если пользователь участвует в другой роли с разрешенными правами на объект или права указаны явно.

Если выбрать строку с объектом таблицы или вьюшки, то внизу окна становится доступной кнопка Columns. Допустим, что ты выбрал таблицу и нажал эту кнопку. Появиться окно, в котором можно настроить доступ к отдельным колонкам таблицы.



Настройка прав доступа на колонки таблицы

Это действительно супер возможность, потому что некоторые колонки, отвечающие за целостность базы не должны изменяться пользователями, и тем более хакерами. На такие колонки лучше запретить операцию изменения (UPDATE) и если есть возможность то даже просмотр (SELECT).

Индивидуализм

Роли очень удобны, когда нужно объединить схожих пользователей. Например, бухгалтерам требуется доступ к финансовым таблицам, которых может быть достаточно много. Выдавать права каждому бухгалтеру в отдельности накладно. Намного проще создать роль для бухгалтера, выдать ей права и потом включить все учетные записи бухгалтерии в эту роль.

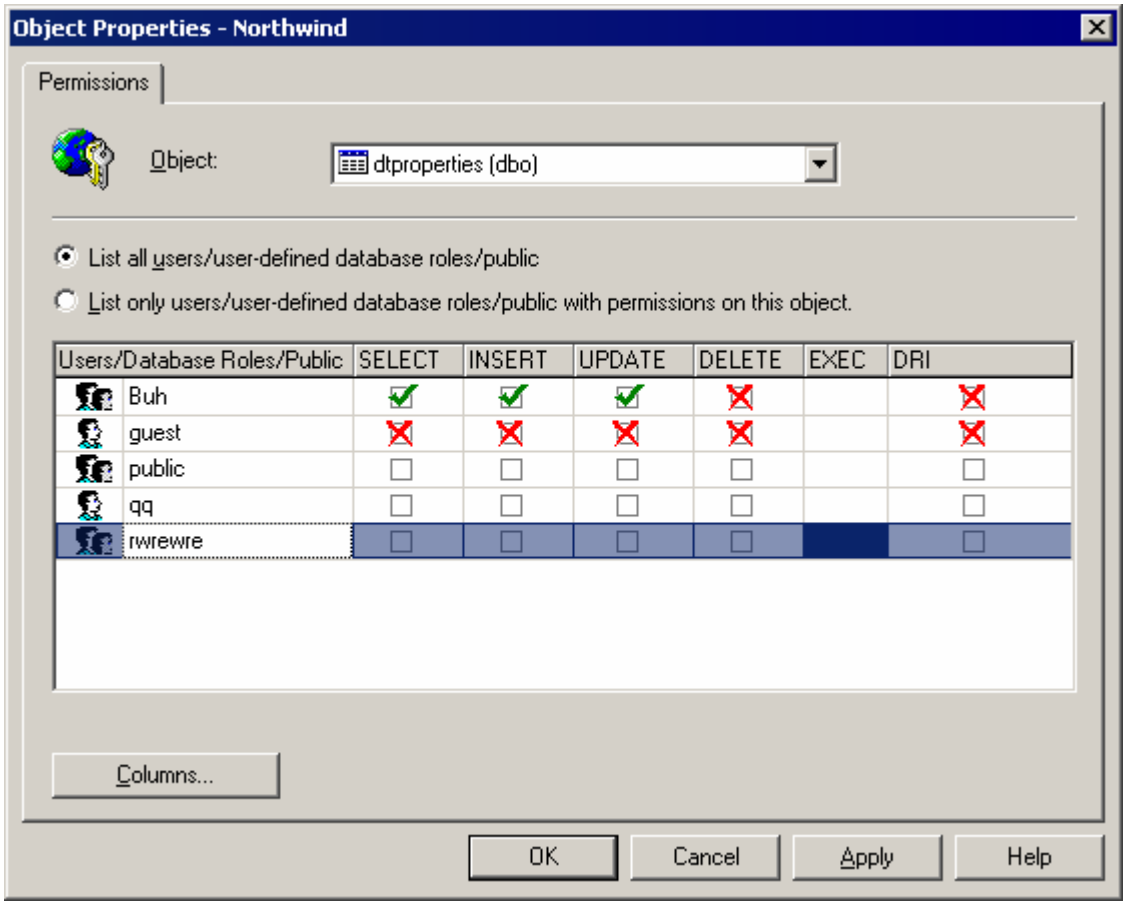
Но бывают случаи, когда права должны быть уникальными для пользователя или помимо тех прав, которые дает роль, нужно дать еще и дополнительные права. Например, один из бухгалтеров может захотеть иметь доступ к таблицам из отдела кадров. Это нормальная ситуация. Вот тут уже заводить отдельную роль нет смысла, а лучше добавить этому человеку права напрямую.

Мы специально рассмотрели сначала роли, чтобы привыкнуть к ним. Просто многие заводят одну запись для бухгалтеров, одну для экономистов и т.д. В этом случае, кучки народа ломятся на сервер через одну запись и контролировать, кто и что сделал невозможно. Индивидуальные права нужно использовать только там, где нужно, а каждый пользователь должен иметь свою запись.

Некоторые считают, что бос должен видеть все, он же босс!!! Это верно, если начальник хороший спец в ИТ и можно быть уверенным, что не навредит. Если он чайник и по случайности уничтожит данные, то виноватым будешь ты, поэтому не давай права sa даже босу.

Права на таблицы

Давай посмотрим, как можно давать права на определенные объекты. Для начала посмотрим таблицы. Выбери в дереве объектов ветку Databases/Northwind/Tables и в правой части будет показан список всех таблиц. Щелкни по любой таблице правой кнопкой и в появившемся меню выбери All tasks/Manage permissions. Перед нами открывается окно настройки прав. Ничего не напоминает это окошко? Я тоже думаю, что дежавю какое-то :). Окно похоже на распределение прав ролей, только вместо списка объектов стоит список пользователей. Объект и так ясен – это та таблица, по которой мы щелкали, и ее имя виднеться в выпадающем списке вверху окна. Теперь нам остается только указать права для этого объекта различным пользователям.



Настройка прав на таблицу

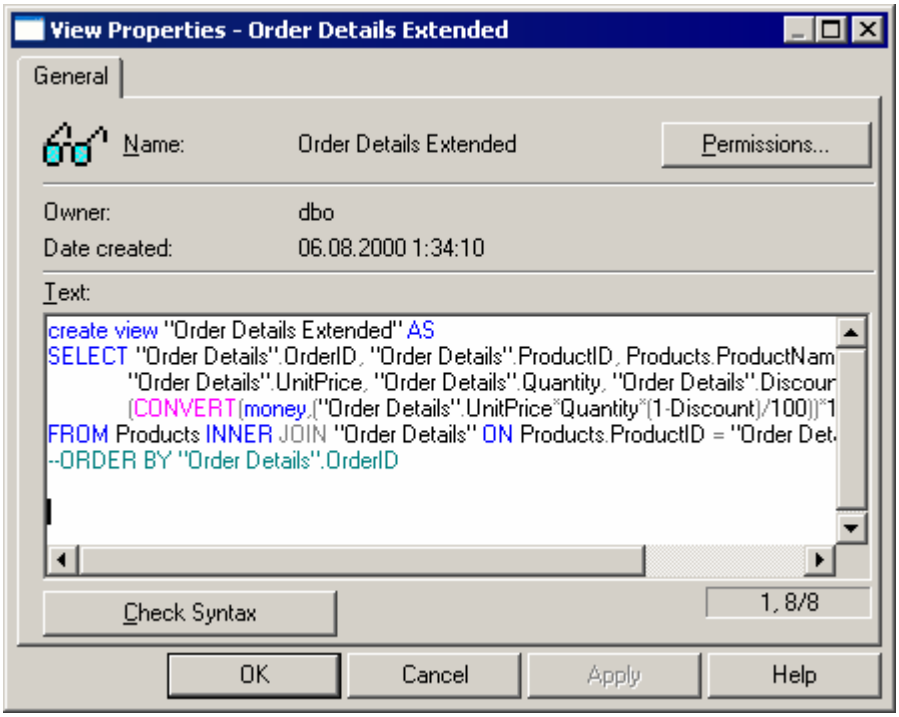
Список прав уже знаком. Это все те же просмотр, обновление, добавление, удаление, выполнение и управление. Если нажать на кнопку Columns, то перед нами откроется окно настройки прав доступа объекта на уровне полей таблицы для выбранного пользователя.

Вьюшки

Допустим, что у нас есть две таблицы. В одной из них хранится список работников фирмы, а в другой находится информация о количестве отработанных часов в месяц и полученной заработной плате (белой и черной). Допустим, что к вам приходит налоговая инспекция и говорит, а покажите нам зарплату работников! А все ли вы налоги заплатили? Какие нужно выполнить действия?

Первое предложение поступило из третьего ряда – создать нового пользователя, которому разрешен доступ на чтение (изменение и другие права налоговой не нужны) таблиц со списком работников и зарплаты. При этом нужно не забыть закрыть колонку с черным налом, иначе босс даст по мягкому месту в районе копчика. В принципе, решение верное, но абсолютно не эффективное.

Лучшим вариантом может быть создание вьюшки (View). Вьюшка – это просто запрос на языке SQL, который выбирает данные, а в базе данных она выглядит как таблица и работа с ней происходит также. Из вьюшки можно выбирать данные SQL запросами и также назначать права. Получается, что будет выполняться запрос к запросу.



Окно создания вьюшки

Для создания вьюшки можно выполнить примерно следующий запрос:

CREATE VIEW Зарплата AS

SELECT разрешенные для налоговой поля

FROM Работники, Доходы

WHERE навести связи

Теперь в ветке Databases/Northwind/Views. Появился новый объект «Зарплата». Если щелкнуть по нему правой кнопкой и в появившемся меню выбрать All tasks/Manage permissions, то перед нами откроется окно настройки прав, как для таблиц. Настраиваем доступ для доступа налоговой и сохраняем. Чтобы просмотреть содержимое вьюшки нужно выполнить запрос:

```
SELECT *
FROM Зарплата
```

Как видишь, обращение происходит как к простой таблице, и налоговая тоже будет думать, что она видит реальные данные, хотя в результате такого запроса будет находиться только то, что нам нужно.

В реальной жизни налоговую полицию так не обманешь, потому что там сидят далеко не лохи. Но на этом примере видно, что вьюшка может оказаться отличным методом обеспечения безопасности. Мы можем отображать пользователям только те данные, которые им нужны и ничего больше. При этом в наших руках остаются все инструменты по управлению правами доступа на вьюшку, не затрагивая права доступа на сами таблицы.

Таким образом, с помощью разных вьюшек к одним и тем же таблицам экономисты могут видеть одни данные, бухгалтерия другие, а отдел кадров третьи. Если нужно показать какую-то дополнительную колонку, то просто добавляем в запрос вьюшки и все. Никаких прав изменять уже не надо будет.

Системные вьюшки

В каждой базе данных могут быть системные вьюшки, которые создаются сервером автоматически. Не советую разрешать к ним доступ, потому что они могут показать что-нибудь лишнее, что поможет хакеру поднять свои права или просто испортить данные. Системные вьюшки начинаются с префикса sys и в колонке Type списка светиться надпись System.

Процедуры и функции

Современные сервера баз данных поддерживают очень удобную вещицу – хранимые процедуры и функции. Это код на языке PL/SQL или Transact-SQL (зависит от базы), который выполняется прямо на сервере баз данных. Через такие процедуры можно выполнять какие-либо действия на сервере или просто выбирать данные, как во вьюшке. Каждой процедуре можно назначать свои права.

При рассмотрении ролей мы уже видели процедуры в списке объектов, на которые можно назначать права и в этих строчках доступна только колонка EXEC (выполнение), потому что процедуры можно только выполнять.

Хранимые процедуры и функции расположены внутри определенной базы. Чтобы увидеть процедуры, например, базы данных Northwind, выбери ветку Databases/Northwind/Stored Procedures. Здесь полно системных процедур, имена которых начинаются с префикса dt_, а в колонке Type светится надпись System. К таким процедурам лучше не давать доступ никому, если нет особой надобности. Функции можно увидеть в ветке Databases/Northwind/User defined function.

Чтобы изменить права доступа процедуры и функции нужно щелкнуть по ее имени правой кнопкой и в появившемся меню выбрать All tasks/Manage permissions. Перед нами откроется окно, как при назначении прав для вьюшек, но для процедур можно изменять только колонку EXEC, а для функций EXEC и DRI.

Политика прав

Некоторые администраторы любят назначать права, взяв за основу какую-то существующую роль, например, public. Это не верно, потому что в этой роли может быть такие права, которые абсолютно не нужны пользователям. Старайтесь назначать права с самого нуля.

Я всегда начинаю новую роль с голого листа и даю только самый необходимый минимум. Если пользователи просят больше прав, и они действительно нужны, то я

повышаю права. Если действовать наоборот и изначально разрешать все, то где гарантия, что в последствии ненужные и в то же время опасные права будут убраны.

Еще одна проблема понижения разрешений кроется в привычке. Пользователи могут привыкнуть, что им многое разрешено и потом запрет будет происходить с большим скандалом. Никто не любит, когда их права ущемляют дверью.

Таблицы/базы

Базы данных хранят свои настройки и секретные параметры не в реестре и не в отдельных файлах, а в системных таблицах/базах данных. Они ничем не отличаются от других объектов базы и на них так же могут назначаться права. Ни в коем случае не разрешай пользователям права на доступ к этим таблицам без особой надобности.

В SQL сервере особо важные системные данные хранятся в базах данных master и msdb. Именно эти базы данных необходимо защищать. В Oracle дело обстоит иначе, потому что там каждая база данных существует как отдельный объект и системные таблицы находятся вперемешку с пользовательскими.

Практически все сервера баз данных предлагают (или без спроса) установить тестовые базы данных, которые могут использоваться для обучения или тестирования системы. Если у тебя стоят такие, то обязательно сейчас же удали, потому что на эти базы устанавливается публичный доступ. Если взломщик будет знать имена или параметры любого реально существующего объекта в системе, то это может упростить его задачу.

Приконнектившись к тестовой базе, можно уже выполнять какие-то команды от имени сервера и навредить ОС или рабочей базе данных. Ничего лишнего в системе не должно быть. Тем более что на такие таблицы/базы данных стоят достаточно высокие права даже для гостя.

Итого

Несмотря на то, что в качестве примера использовалась MS SQL Server, понятия прав, ролей и аутентификации есть практически во всех базах данных. Зная все рассмотренные нами правила, тебе осталось только узнать специфику их использования в твоей базе и ты в безопасности.