ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ, ИНФОРМАТИЗАЦИИ И ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ.

КУРСОВАЯ РАБОТА

« С З Б Д »

ТЕМА: “ ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ В СОВРЕМЕННЫХ СИСТЕМАХ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ”

Выполнил: Гр. 232-10 АХу
АДИЛОВ О.

Принял: ГУЛЯМОВ Ш.

ТАШКЕНТ-2013.

Содержания:

Введение

В современных условиях любая деятельность сопряжена с оперированием большими объемами информации, которое производится широким кругом лиц. Защита данных от несанкционированного доступа является одной из приоритетных задач при проектировании любой информационной системы. Следствием возросшего в последнее время значения информации стали высокие требования к конфиденциальности данных. Системы управления базами данных, в особенности реляционные СУБД, стали доминирующим инструментом в этой области. Обеспечение информационной безопасности СУБД приобретает решающее значение при выборе конкретного средства обеспечения необходимого уровня безопасности организации в целом.

ГЛАВА 1. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ В СОВРЕМЕННЫХ СИСТЕМАХ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ.

В современных условиях любая деятельность сопряжена с оперированием большими объемами информации, которое производится широким кругом лиц. Защита данных от несанкционированного доступа является одной из приоритетных задач при проектировании любой информационной системы. Следствием возросшего в последнее время значения информации стали высокие требования к конфиденциальности данных. Системы управления базами данных, в особенности реляционные СУБД, стали доминирующим инструментом в этой области. Обеспечение информационной безопасности СУБД приобретает решающее значение при выборе конкретного средства обеспечения необходимого уровня безопасности организации в целом.

Для СУБД важны три основных аспекта информационной безопасности — конфиденциальность, целостность и доступность. Темой настоящей статьи является первый из них — средства защиты от несанкционированного доступа к информации. Общая идея защиты базы данных состоит в следовании рекомендациям, сформулированным для класса безопасности С2 в «Критериях оценки надежных компьютерных систем»[1].

Политика безопасности определяется администратором данных. Однако решения защиты данных не должны быть ограничены только рамками СУБД. Абсолютная защита данных практически не реализуема, поэтому обычно довольствуются относительной защитой информации — гарантированно защищают ее на тот период времени, пока несанкционированный доступ к ней влечет какие-либо последствия. Разграничение доступа к данным также описывается в базе данных посредством ограничений, и информация об этом хранится в ее системном каталоге. Иногда дополнительная информация может быть запрошена из

операционных систем, в окружении которых работают сервер баз данных и клиент, обращающийся к серверу баз данных.

1.1 Некоторые термины.

Конфиденциальная информация (sensitive information) — информация, которая требует защиты.

Доступ к информации (access to information) — ознакомление с информацией, ее обработка (в частности, копирование), модификация, уничтожение.

Субъект доступа (access subject) — лицо или процесс, действия которого регламентируются правилами разграничения доступа.

Объект доступа (access object) — единица информации автоматизированной системы, доступ к которой регламентируется правилами разграничения доступа. Объектами доступа (контроля) в СУБД является практически все, что содержит конечную информацию: таблицы (базовые или виртуальные), представления, а также более мелкие элементы данных: столбцы и строки таблиц и даже поля строк (значения). Таблицы базы данных и представления имеют владельца или создателя. Их объединяет еще и то, что все они для конечного пользователя представляются как таблицы, то есть как нечто именованное, содержащее информацию в виде множества строк (записей) одинаковой структуры. Строки таблиц разбиты на поля именованными столбцами.

Правила разграничения доступа (security policy) — совокупность правил, регламентирующих права субъектов доступа к объектам доступа.

Санкционированный доступ (authorized access to information) — доступ к информации, который не нарушает правил разграничения доступа.

Несанкционированный доступ (unauthorized access to information) — доступ к информации, который нарушает правила разграничения доступа с использованием штатных средств, предоставляемых средствами вычислительной техники или автоматизированными системами.

Идентификатор доступа (access identifier) — уникальный признак объекта или субъекта доступа.

Идентификация (identification) — присвоение объектам и субъектам доступа идентификатора и (или) сравнение предъявляемого идентификатора с перечнем присвоенных идентификаторов.

Пароль (password) — идентификатор субъекта, который является его секретом.

Аутентификация (authentification) — проверка принадлежности субъекту доступа предъявленного им идентификатора, подтверждение подлинности.

В СУБД на этапе подключения к БД производится идентификация и проверка подлинности пользователей. В дальнейшем пользователь или процесс получает доступ к данным согласно его набору полномочий. В случае разрыва соединения пользователя с базой данных текущая транзакция откатывается, и при восстановлении соединения требуется повторная идентификация пользователя и проверка его полномочий.

Уровень полномочий субъекта доступа (subject privilege) — совокупность прав доступа субъекта доступа (для краткости в дальнейшем мы будем использовать термин «привилегия»).

Нарушитель правил разграничения доступа (security policy violator) — субъект доступа, который осуществляет несанкционированный доступ к информации.

Модель нарушителя правил разграничения доступа (security policy violator model) — абстрактное (формализованное или неформализованное) описание нарушителя правил разграничения доступа.

Целостность информации (information integrity) — способность средства вычислительной техники (в рассматриваемом случае — информационной системы в целом) обеспечить неизменность информации в условиях случайного и (или) преднамеренного искажения (разрушения).

Метка конфиденциальности (sensitivity label) — элемент информации, характеризующий конфиденциальность объекта.

Многоуровневая защита (multilevel secure) — защита, обеспечивающая разграничение доступа субъектов с различными правами доступа к объектам различных уровней конфиденциальности.

1.2 Пользователи СУБД.

Пользователей СУБД можно разделить на три группы:

1. Прикладные программисты — отвечают за создание программ, использующих базу данных.

   В смысле защиты данных программист может быть как пользователем, имеющим привилегии создания объектов данных и манипулирования ими, так и пользователем, имеющим привилегии только манипулирования данными.

2. Конечные пользователи базы данных — работают с БД непосредственно через терминал или рабочую станцию. Как правило, конечные пользователи имеют строго ограниченный набор привилегий манипулирования данными. Этот набор может определяться при конфигурировании интерфейса конечного пользователя и не изменяться. Политику безопасности в данном случае определяет администратор безопасности или администратор базы данных (если это одно и то же должностное лицо).

3. Администраторы баз данных — образуют особую категорию пользователей СУБД. Они создают сами базы данных, осуществляют технический контроль функционирования СУБД, обеспечивают необходимое быстродействие системы. В обязанности администратора, кроме того, входит обеспечение пользователям доступа к необходимым им данным, а также написание (или оказание помощи в определении) необходимых пользователю внешних представлений данных. Администратор определяет правила безопасности и целостности данных.

1.3 Дискреционная защита.

Всовременных СУБД достаточно развиты средства дискреционной защиты.

Дискреционное управление доступам (discretionary access control) — разграничение доступа между поименованными субъектами и поименованными объектами. Субъект с определенным правом доступа может передать это право любому другому субъекту.

Дискреционная защита является многоуровневой логической защитой.

Логическая защита в СУБД представляет собой набор привилегий или ролей по отношению к защищаемому объекту. К логической защите можно отнести и владение таблицей (представлением). Владелец таблицы может изменять (расширять, отнимать, ограничивать доступ) набор привилегий (логическую защиту) . Данные о логической защите находятся в системных таблицах базы данных и отделены от защищаемых объектов (от таблиц или представлений).

Информация о зарегистрированных пользователях базы данных хранится в ее системном каталоге. Современные СУБД не имеют общего синтаксиса SQL-предложения соединения с базой данных, так как их собственный синтаксис сложился раньше, чем стандарт ISO. Тем не менее часто таким ключевым предложением является CONNECT. Ниже приведен синтаксис данного предложения для Oracle и IBM DB2 соответственно:

CONNECT [[logon] [AS {SYSOPER|SYSDBA}]] пользователь/пароль[@база_данных]

CONNECT TO база_данных USER пользователь USING пароль

В данных предложениях отражен необходимый набор атрибутов, а также показано различие синтаксиса. Формат атрибута база_данных, как правило, определяется производителем СУБД, так же как и имя

пользователя, имеющего по умолчанию системные привилегии (SYSDBA/SYSOPER в случае Oracle).

Соединение с системой не идентифицированных пользователей и пользователей, подлинность идентификации которых при аутентификации не подтвердилась, исключается. В процессе сеанса работы пользователя (от удачного прохождения идентификации и аутентификации до отсоединения от системы) все его действия непосредственно связываются с результатом идентификации. Отсоединение пользователя может быть как нормальным (операция DISCONNECT), так и насильственным (исходящим от пользователя-администратора, например в случае удаления пользователя или при аварийном обрыве канала связи клиента и сервера). Во втором случае пользователь будет проинформирован об этом, и все его действия аннулируются до последней фиксации изменений, произведенных им в таблицах базы данных. В любом случае на время сеанса работы идентифицированный пользователь будет субъектом доступа для средств защиты информации от несанкционированного доступа (далее — СЗИ НСД) СУБД.

Следуя технологии открытых систем, субъект доступа может обращаться посредством СУБД к базе данных только из программ, поставляемых в дистрибутиве или подготовленных им самим, и только с помощью штатных средств системы.

Все субъекты контроля системы хранятся в таблице полномочий системы и разделены для системы на ряд категорий, например CONNECT, RESOURCE и DBA. Набор таких категорий определяется производителем СУБД. Мы не случайно предлагаем указанный порядок рассмотрения — именно так происходит нарастание возможностей (полномочий) для каждого отдельного вида подключения:

- CONNECT — конечные пользователи. По умолчанию им разрешено только соединение с базой данных и выполнение запросов к

данным, все их действия регламентированы выданными им привилегиями;

- RESOURCE — привилегированные пользователи, обладающие правом создания собственных объектов в базе данных (таблиц, представлений, синонимов, хранимых процедур). Пользователь — владелец объекта обладает полным набором привилегий для управления данным объектом;
- DBA — категория администраторов базы данных. Включает возможности обеих предыдущих категорий, а также возможность вводить (удалять) в систему (из системы) субъекты защиты или изменять их категорию.

Следует особо отметить, что в некоторых реализациях административные действия также разделены, что обусловливает наличие дополнительных категорий. Так, в Oracle пользователь с именем DBA является администратором сервера баз данных, а не одной-единственной базы данных. В СУБД «Линтер» компании РЕЛЭКС понятие администратора сервера баз данных отсутствует, а наличествует только понятие администратора конкретной базы данных. В IBM DB2 существует ряд категорий администраторов: SYSADM (наивысший уровень; системный администратор, обладающий всеми привилегиями); DBADM (администратор базы данных, обладающий всем набором привилегий в рамках конкретной базы данных). Привилегии управления сервером баз данных имеются у пользователей с именами SYSCTRL (наивысший уровень полномочий управления системой, который применяется только к операциям, влияющим на системные ресурсы; непосредственный доступ к данным запрещен, разрешены операции создания, модификации, удаления базы данных, перевод базы данных или экземпляра (instance) в пассивное состояние (quiesce), создание и удаление табличных пространств), SYSMAINT (второй уровень полномочий управления системой, включающий все операции поддержки работоспособности экземпляра (instance); непосредственный

доступ к данным этому пользователю запрещен, разрешены операции изменения конфигурационных файлов базы данных, резервное копирование базы данных и табличных пространств, зеркалирование базы данных). Для каждой административной операции в IBM DB2 определен необходимый набор административных категорий, к которым должен принадлежать пользователь, выполняющий тот или иной запрос администрирования. Так, выполнять операции назначения привилегий пользователям может SYSADM или DBADM, а для того чтобы создать объект данных, пользователь должен обладать привилегией CREATETAB.

Администратор каждой базы занимается созданием круга возможных пользователей создаваемой им БД и разграничением полномочий этих пользователей. Данные о разграничениях располагаются в системном каталоге БД. Очевидно, что данная информация может быть использована для несанкционированного доступа и поэтому подлежит защите. Защита этих данных осуществляется средствами самой СУБД.

СУБД позволяет зарегистрировать пользователя и хранить информацию о его уникальном идентификаторе. Например, подсистема безопасности Oracle позволяет создавать пользователей базы данных посредством предложения:

CREATE USER IDENTIFIED пользователь BY пароль

Подсистема безопасности IBM DB2 может использовать идентификаторы пользователей операционной системы; ее синтаксис SQL не содержит предложения, аналогичного предложению CREATE USER. Microsoft SQL Server может использовать аутентификацию как базы данных, так и операционной системы. Но мы не станем здесь обсуждать достоинства и недостатки выбранных производителями способов аутентификации — все они позволяют строить корректные схемы определения подлинности пользователей. Использование дополнительных средств аутентификации в рамках информационной системы не запрещается.

Набор привилегий можно определить для конкретного зарегистрированного пользователя или для группы пользователей (это могут быть собственно группы пользователей, роли и т.п.). Объектом защиты может являться таблица, представление, хранимая процедура

и т.д. (подробный список объектов защиты имеется в документации к используемой СУБД). Субъектом защиты может быть пользователь, группа пользователей или роль, а также хранимая процедура, если такое предусматривается используемой реализацией. Если из используемой реализации следует, что хранимая процедура имеет «двойной статус» (она и объект защиты, и субъект защиты), то нужно очень внимательно рассмотреть возможные модели нарушителей разграничения прав доступа и предотвратить эти нарушения, построив, по возможности, соответствующую систему защиты.

При использовании хранимых процедур следует обращать особое внимание на то, от имени какого пользователя выполняется данная хранимая процедура в каждом конкретном случае. Так, в Oracle до недавнего времени хранимые процедуры выполнялись от имени владельца хранимой процедуры, а не от имени пользователя, выполнившего ее вызов. Текущая версия Oracle предоставляет возможность указать, под чьим именем будет выполняться вызванная хранимая процедура, пользователь же должен иметь только привилегию EXECUTE для данной процедуры. В «Линтер», например, выполнение хранимых процедур всегда происходит от имени пользователя, вызвавшего процедуру.

Привилегии конкретному пользователю могут быть назначены администратором явно и неявно, например через роль. Роль — это еще один возможный именованный носитель привилегий. С ролью не ассоциируют перечень допустимых пользователей — вместо этого роли защищают паролями, если, конечно, такая возможность поддерживается производителем СУБД. Роли удобно использовать, когда тот или иной набор привилегий необходимо выдать (или отобрать) группе пользователей. С

одной стороны, это облегчает администратору управление привилегиями, с другой — вносит определенный порядок в случае необходимости изменить набор привилегий для группы пользователей сразу. Нужно особо отметить, что при выполнении хранимых процедур и интерактивных запросов может существовать зависимость набора привилегий пользователя от того, как они были получены: явно или через роль. Имеют место и реализации, например в Oracle, где в хранимых процедурах используются привилегии, полученные пользователем явно. Если используемая вами реализация обладает подобным свойством, то изменение привилегий у группы пользователей следует реализовать как набор команд или как административную процедуру (в зависимости от предпочтений администратора).

Предложения управления привилегиями:

- назначение привилегии:

GRANT привилегия [ON объект] TO субъект [WITH GRANT OPTION]

- отмена привилегии:

REVOKE привилегия [ON объект] FROM субъект

Если субъект=пользователь, то привилегия назначается ему явно. Если субъект=роль, то для управления привилегиями используются соответственно:

GRANT ROLE имя_роли [ON объект] TO субъект [WITH GRANT OPTION]

REVOKE ROLE имя_роли [ON объект] FROM субъект

Назначение привилегии всем пользователям системы осуществляется следующим образом:

GRANT привилегия [ON объект] TO PUBLIC

В этом случае каждый новый созданный пользователь автоматически получит такую привилегию. Отмена привилегии осуществляется так:

REVOKE привилегия [ON объект] FROM PUBLIC

Необходимо иметь в виду, что некоторые реализации, например IBM DB2, используют группы пользователей, определенные в операционной системе. Поэтому следует обращать внимание на особенности реализации аналогов ролей в СУБД. Нужно выяснить, содержит ли реализация SQL-предложения вида:

CREATE ROLE имя_роли

DROP ROLE имя_роли

При управлении доступом к таблицам и представлениям набор привилегий в реализации СУБД определяется производителем.

Привилегии выборки и модификации данных:

SELECT — привилегия на выборку данных;

INSERT — привилегия на добавление данных;

DELETE — привилегия на удаление данных;

UPDATE — привилегия на обновление данных (можно указать определенные столбцы, разрешенные для обновления).

Привилегии изменения структуры таблиц:

ALTER — изменение физической/логической структуры базовой таблицы (изменение размеров и числа файлов таблицы, введение дополнительного столбца и т.п.);

INDEX — создание/удаление индексов на столбцы базовой таблицы;

ALL — все возможные действия над таблицей.

В реализациях могут присутствовать другие разновидности привилегий, например: CONTROL (IBM DB2) — комплексная привилегия управления структурой таблицы, REFERENCES — привилегия создания внешних ключей, RUNSTAT — выполнение сбора статистической информации по таблице и другие.

Однако дискреционная защита является довольно слабой, так как доступ ограничивается только к именованным объектам, а не собственно к хранящимся данным. В случае реализации информационной системы с использованием реляционной СУБД объектом будет, например, именованное

отношение (то есть таблица), а субъектом — зарегистрированный пользователь. В этом случае нельзя в полном объеме ограничить доступ только к части информации, хранящейся в таблице. Частично проблему ограничения доступа к информации решают представления и использование хранимых процедур, которые реализуют тот или иной набор бизнес-действий.

Представление (view) — это сформированная выборка кортежей, хранящихся в таблице (таблицах). К представлению можно обращаться точно так же, как и к таблицам, за исключением операций модификации данных, поскольку некоторые типы представлений являются немодифицируемыми. Часто в реализациях view хранится как текст, описывающий запрос выборки, а не собственно выборка данных; выборка же создается динамически на момент выполнения предложения SQL, использующего view. Но разграничить доступ, например, к двум документам, которые удовлетворяют одному и тому же условию выборки, уже нельзя. Это связано с тем, что даже если ввести отдельный атрибут, который будет хранить информацию о метке конфиденциальности документа, то средствами SQL можно будет получить выборку данных без учета атрибута данной метки. Фактически это означает, что либо сам сервер баз данных должен предоставить более высокий уровень защиты информации, либо придется реализовать данный уровень защиты информации с помощью жесткого ограничения операций, которые пользователь может выполнить посредством SQL. На некотором уровне такое разграничение можно реализовать с помощью хранимых процедур, но не полностью — в том смысле, что само ядро СУБД позволяет разорвать связь «защищаемый объект Ц метка конфиденциальности».

1.4 Мандатная защита.

Средства мандатной защиты предоставляются специальными (trusted) версиями СУБД.

Мандатное управление доступом (mandatory access control) — это разграничение доступа субъектов к объектам данных, основанное на

характеризуемой меткой конфиденциальности информации, которая содержится в объектах, и на официальном разрешении (допуске) субъектов обращаться к информации такого уровня конфиденциальности.

Для чего же нужна мандатная защита? Средства произвольного управления доступом характерны для уровня безопасности С. Как правило, их, в принципе, вполне достаточно для подавляющего большинства коммерческих приложений. Тем не менее они не решают одной весьма важной задачи — задачи слежения за передачей информации. Средства произвольного управления доступом не могут помешать авторизованному пользователю законным образом получить секретную информацию и затем сделать ее доступной для других, неавторизованных, пользователей. Нетрудно понять, почему это так. При произвольном управлении доступом привилегии существуют отдельно от данных (в случае реляционных СУБД — отдельно от строк реляционных таблиц), в результате чего данные оказываются «обезличенными» и ничто не мешает передать их кому угодно даже средствами самой СУБД; для этого нужно лишь получить доступ к таблице или представлению.

Физическая защита СУБД главным образом характеризует данные (их принадлежность, важность, представительность и пр.). Это в основном метки безопасности, описывающие группу принадлежности и уровни конфиденциальности и ценности данных объекта (таблицы, столбца, строки или поля). Метки безопасности (физическая защита) неизменны на всем протяжении существования объекта защиты (они уничтожаются только вместе с ним) и территориально (на диске) располагаются вместе с защищаемыми данными, а не в системном каталоге, как это происходит при логической защите.

СУБД не дает проигнорировать метки конфиденциальности при получении доступа к информации. Такие реализации СУБД, как правило, представляют собой комплекс средств как на машине-сервере, так и на машине-клиенте, при этом возможно использование специальной

защищенной версии операционной системы. Кроме разграничения доступа к информации посредством меток конфиденциальности, защищенные СУБД предоставляют средства слежения за доступом субъектов к объектам защиты (аудит).

Использование СУБД с возможностями мандатной защиты позволяет разграничить доступ собственно к данным, хранящимся в информационной системе, от доступа к именованным объектам данных. Единицей защиты в этом случае будет являться, в частности, запись о договоре N, а не таблица или представление, содержащее информацию об этом договоре. Пользователь, который будет пытаться получить доступ к договору, уже никак не сможет обойти метку конфиденциальности. Существуют реализации, позволяющие разграничивать доступ вплоть до конкретного значения конкретного атрибута в конкретной строке конкретной таблицы. Дело не ограничивается одним значением метки конфиденциальности — обычно сама метка представляет собой набор значений, отражающих, например, уровень защищенности устройства, на котором хранится таблица, уровень защищенности самой таблицы, уровень защищенности атрибута и уровень защищенности конкретного кортежа.

За исключением атрибута собственности (логическая защита), разбивающего данные (таблицы) на собственные (принадлежащие данному субъекту) и чужие, физическая защита разбивает данные более тонко. Но можно ли обойтись без физической защиты или, по крайней мере, попытаться, реализовав, например, сложный набор хранимых процедур. В общем-то некоторое подобие такой защиты реализуемо в случае, когда метки добавляются в таблицу в качестве дополнительного атрибута, доступ к таблицам запрещается вообще и ни одно приложение не может выполнить интерактивный SQL-запрос, а только хранимую процедуру и т.п. Ряд реализаций подобного уровня защиты использует вызов набора хранимых процедур с весьма абстрактными (что очень желательно) именами. Система реализации защиты информации в данном случае достаточно сложна и

предполагает определенный уровень доверия к администратору безопасности, так как он имеет право изменять структуру базы данных, а значит, и хранимые процедуры, представления. Физически же администратор безопасности в данном случае не изолирован от управления секретными данными.

Кроме того, защищенные СУБД позволяют разграничить доступ к информационной системе с тех или иных рабочих станций для тех или иных зарегистрированных пользователей, определить режимы работы, наложить ограничения по времени работы тех или иных пользователей с тех или иных рабочих станций. В случае реализации данных опций на прикладном уровне задача, как правило, сводится к созданию сервера приложений, который занимается отслеживанием, «кто и откуда пришел». Отдельный комплекс серверных приложений (обычно — хранимых процедур, если в СУБД отсутствует мандатная защита) обеспечивает аудит.

Рассмотрим мандатную защиту подробнее. В качестве примера возьмем мандатную защиту СУБД «Линтер», которая получила признание в весьма специфическом секторе — силовых структурах, как единственная СУБД, имеющая сертификат по второму классу защиты от несанкционированного доступа, что соответствует классу В3 по американскому национальному стандарту.

Во-первых, все перечисленные объекты (независимо от их иерархии в базе данных) разбиваются здесь на группы принадлежности. Объект может принадлежать только одной из групп (это может быть, например, разбиение по отделам организации). Группы принадлежности напрямую связаны с группами субъектов (см. ниже). Субъект вправе видеть только данные своей группы, если между группами субъектов не установлены отношения доверия.

Во-вторых, все объекты выстроены в иерархию по уровням конфиденциальности и по уровням ценности или важности. Уровень конфиденциальности разбивает объекты по доступности на чтение (и даже на просмотр). Пользователь с более низким уровнем доступа не будет знать

даже о существовании объектов с более высоким уровнем конфиденциальности. Уровень ценности, напротив, разбивает данные (объекты) по важности, ограничивая возможность их удаления и модификации.

В уже упоминавшихся «Критериях оценки надежных компьютерных систем» применительно к системам уровня безопасности В описан механизм меток безопасности, реализованный в рассматриваемых данной статьей СУБД.

Метка объекта включает следующее:

1. Группа субъекта, который внес данный объект.
2. Уровень доступа на чтение — RAL (Read Access Level).
3. Уровень доступа на запись — WAL (Write Access Level).

Метка субъекта выглядит аналогично:

1. Группа, к которой принадлежит субъект.
2. RAL-уровень субъекта, который представляет собой максимальный RAL-уровень доступной субъекту информации.
3. WAL-уровень субъекта, то есть минимальный RAL-уровень объекта, который может быть создан этим субъектом.

Все пользователи базы данных считаются разбитыми на непересекающиеся группы. Группа описывает область доступных пользователю данных. Для каждой группы существует администратор группы (уровень DBA для группы), созданный администратором системы. При этом пользователи одной группы не видят данных, принадлежащих пользователям другой группы. В этом плане у СУБД «Линтер» имеется особенность: в системе реализовано такое понятие, как «уровень доверия между группами». При этом уровни доверия не могут быть вложенными. Группа представляет собой числовое значение в диапазоне [1-250]. Группа 0 — группа администратора системы. Только администратор системы может создать пользователя в группе, отличной от своей. Все данные, созданные от имени пользователя, помечаются его группой.

Уровни доступа вводятся для проверки прав на осуществление чтения-записи информации. Вводятся следующие уровни доступа:

1. Для пользователя (субъекта):

- RAL — уровень доступа; пользователь может получать (читать) информацию, RAL-уровень которой не выше его собственного уровня доступа;
- WAL — уровень доверия на понижение уровня конфиденциальности; пользователь не может вносить информацию с уровнем доступа (RAL-уровнем) более низким, чем данный WAL-уровень пользователя. Иными словами, пользователь не может сделать доступную ему информацию менее конфиденциальной, чем указано в данном параметре.

1. Для информации:

- RAL — уровень чтения; пользователь может получать (читать) информацию, RAL-уровень которой не выше его собственного RAL-уровня (может читать менее конфиденциальные данные);
- WAL — уровень ценности или уровень доступа на запись (модификацию, удаление); пользователь может модифицировать (удалять) информацию, WAL-уровень которой не выше его RAL-уровня.

Создать пользователя с произвольными уровнями может только администратор системы. Остальные администраторы (DBA) могут создавать пользователей (или изменять уровень пользователям) лишь в пределах отведенных им уровней. Пользователь может принудительно пометить вводимые данные, указав в списке атрибутов уровни доступа для соответствующих записей и полей (при выполнении операторов INSERT или UPDATE). По умолчанию вносимые данные наследуют уровни пользователя, вносящего/изменяющего данные. Защищаемые объекты: пользователи, таблицы, столбцы, записи (вносится при выполнении INSERT), поля записей

(изменяются при выполнении UPDATE). Уровни, как и группы, нельзя использовать в случае, если они не созданы специальными запросами.

Конфигурация, к которой имеет доступ хотя бы один программист, не может считаться безопасной. Поэтому обеспечение информационной безопасности баз данных — дело весьма сложное, и во многом вследствие самой природы реляционных СУБД.

Помимо систематического применения арсенала средств, описанных выше, необходимо использовать административные и процедурные меры, в частности регулярное изменение паролей пользователей, предотвращение доступа к физическим носителям информации и т.п.

ГЛАВА 2. РАЗРАБОТКЕ МЕХАНИЗМА МАНДАТНОГО УПРАВЛЕНИЯ ДОСТУПОМ В СУБД MY SQL НА ОСНОВЕ SELINUX.

- Оне обходимости исследований по разработке механизма антикризисного управления социально-экономическим развитием муниципальных образований региона самаруха виктор иванович, гуляева людмила валерьевна
- концепция интеллектуального управления технологическими процессами грузового порта на основе имитационных моделей проталинский олег мирославович, ханова анна алексеевна
- формирование механизмов антикризисного управления на основе разработки стратегии развития предприятия в кризисной ситуации никулина тамара николаевна
- организационно-экономические основы формирования механизма эффективного управления фирмой в сфере медицинских услуг колесникова и. С.
- концептуальная модель экономического механизма антикризисного управления торговой организацией коношенко л.а.

2.1 Разработке механизма мандатного управления доступом в СУБД My SQL на основе SELinux.

Рассматривается подход к разработке механизма управления доступом в системе управления базами данных (СУБД) MySQL на основе SELinux.

В СУБД MySQL [1] используется дискреционный механизм управления доступом, реализованный следующим образом. Имеется база данных (БД) mysql, которая содержит таблицы, где хранится различная служебная информация. Записи, отражающие права доступа субъектов к сущностям, находятся в следующих таблицах:

— user;

— db;

— host;

— tables_priv;

— columns_priv;

— procs_priv.

Таблица user хранит глобальные права доступа субъектов к сущностям. Глобальные права распространяются на все сущности СУБД. Таблицы db и host определяют права для БД. В остальных таблицах находятся права, определяющие доступ субъекта к таким сущностям, как таблицы и столбцы.

Процесс управления доступом начинается с проверки наличия глобальных прав. Если они присутствуют, то доступ разрешается. В случае отсутствия каких-либо прав у субъекта в таблице user последовательно просматриваются оставшиеся таблицы.

В СУБД MySQL отсутствуют механизмы мандатного управления доступом. В то же время в известных СУБД (например, Oracle) имеется возможность реализации политики мандатного управления доступом. В СУБД MySQL это может быть реализовано с использованием механизма безопасности SELinux [2].

В системе SELinux политика безопасности управления доступом компьютерной системы (КС) задаётся набором правил, описывающих права доступа субъектов к сущностям на основе их контекста безопасности, в виде единого конфигурационного файла или набора модулей [3, 4]. Под контекстом безопасности понимается набор атрибутов, связанных с сущностью КС и имеющих вид user : role : type[: level], где

— user — атрибут-пользователь в системе SELinux, ассоциированный с одним или более пользователем КС;

— role — атрибут-роль в системе SELinux, ассоциированная с одним или более типом, к которым пользователь SELinux имеет доступ;

— type — атрибут-тип в системе SELinux, определяющий возможные виды доступа сущностей данного типа при использовании мандатного механизма Type Enforcement;

— level — атрибут-уровень безопасности в системе SELinux при использовании мандатных механизмов Multy-Level Security или Multy-Category Security.

Система SELinux состоит из следующих частей:

1) Object Manager (OM) — служит посредником при принятии решения о разрешении/запрете доступа субъекта к объекту;

2) Access Vector Cache (AVC)—необходим для оптимизации работы системы SELinux;

3) сервер SELinux — применяется для создания ответа на основе запроса и политики SELinux.

При попытке субъекта получить доступ к сущности OM составляет запрос и отправляет его к AVC. AVC принимает запрос от OM, и если ранее поступал такой запрос, то AVC находит его в своей базе и отправляет ответ OM, иначе запрос перенаправляется серверу SELinux. Сервер SELinux при поступлении запроса находит соответствующее правило в политике безопасности и отправляет ответ AVC, который, в свою очередь, запоминает его и перенаправляет OM.

OM реализован на уровне ядра ОС GNU/Linux, при этом SELinux предоставляет средства (библиотека libselinux), реализующие OM на уровне пользовательского приложения.

С использованием данной возможности для СУБД MySQL (версия 5.5.16) разработан прототип модуля, задающий мандатную политику управления доступом на основе механизма Type Enforcement, и элемент OM, реализующий её на пользовательском уровне, а также определены контексты безопасности для основных сущностей СУБД MySQL.

2.2 Научная статья по специальности "Автоматика. Вычислительная техника".

В СУБД My SQL для присвоения сущностям контекстов безопасности использованы следующие таблицы БД mysql:

— SEDB;

— SETable;

— SEColumn.

В этих таблицах сопоставляются контексты безопасности и объекты DB, Table, Column СУБД My SQL. В служебной таблице user БД mysql создан столбец sec_context, задающий контексты безопасности субъектам. Для задания контекста безопасности записям необходимо в пользовательской таблице создать столбец sec_context.

Одним из типичных способов реализации элемента ОМ является добавление хук-функций в исходный код, содержащий функции управления доступом субъектов к сущностям КС. В СУБД MySQL такими функциями являются:

— bool check_access() —функция, реализующая управление доступом на основе табличных данных user, db, host;

— bool check_grant() —функция, реализующая управление доступом на основе табличных данных table_priv;

— bool check_grant_column() — функция, реализующая управление доступом на основе табличных данных column_priv.

В момент реализации субъектом доступа к сущности выполняется проверка глобальных прав доступа, при этом вызывается функция check_access(), передающая управление хук-функции. Последняя взаимодействует с сервером SELinux и, в зависимости от ответа, выполняет действия по запрету или разрешению доступа.

Таким образом, основными этапами реализации политики мандатного управления доступом в СУБД MySQL на основе механизма SELinux являются:

1) задание контекста безопасности для каждой сущности СУБД MySQL;

2) разработка модуля политики безопасности;

3) создание функций, реализующих элемент ОМ для СУБД MySQL;

4) замена функций, реализующих управление доступом в СУБД MySQL на функции, вызывающие механизмы SELinux.

2.3 Моделях логического управления доступом на основе атрибутов.

Доклад посвящен обзору основных работ по моделям логического управления доступом на основе атрибутов, или, иначе, атрибутного управления доступом (Attribute Based Access Control) в компьютерных системах (КС). При таком виде управления доступом предоставление субъекту права доступа к сущности происходит только в том случае, если значения атрибутов субъектов и сущностей позволяют субъекту предоставить данный доступ к сущности. Как правило, атрибутное управление доступом рассматривается как отдельный вид управления доступом наряду с дискреционным, мандатным и ролевым. Вместе с тем КС с управлением доступом на основе атрибутов могут использовать в качестве последних типы, уровни безопасности и роли, включая в себя соответственно отдельные дискреционные, мандатные и ролевые механизмы. В общем случае механизмы функционирования атрибутного управления доступом характерны для систем с мандатным управлением доступом [1].

Атрибутное управление доступом является новым и перспективным видом политик логического управления доступом и информационными потоками в КС. Большинство работ по моделям атрибутного управления доступом (например, [2]) ориентированы на реализацию или оптимизацию подсистемы управления доступом; в них используются оригинальные определения элементов и механизмов защиты, а используемый математический аппарат часто недостаточен для анализа условий нарушения безопасности КС и формального обоснования методов и требований их защиты. В то же время известны модели атрибутного управления доступом, например [3], исследующие вопросы теоретического анализа безопасности.

Исторически первой моделью атрибутного управления доступом может считаться модель типизированной матрицы доступа (ТМД), в которой с каждым объектом системы ассоциирован атрибут-тип. В настоящее время предложено несколько подходов к управлению доступом на основе атрибутов.

В [4] подробно рассматривается модель с динамической ролью. Предполагается, что в системе имеется некоторое количество ролей с заранее определенными правами. К системе имеют доступ неограниченное количество пользователей, которым необходимо приписывать какие-то роли. При этом роли могут меняться с течением времени. Для определения роли пользователя в текущий момент времени используются значе-

Заключение.

Доклад посвящен обзору основных работ по моделям логического управления доступом на основе атрибутов, или, иначе, атрибутного управления доступом (Attribute Based Access Control) в компьютерных системах (КС). При таком виде управления доступом предоставление субъекту права доступа к сущности происходит только в том случае, если значения атрибутов субъектов и сущностей позволяют субъекту предоставить данный доступ к сущности.

Принимая тот или иной подход к обеспечению безопасности информационных ресурсов, организации либо идут на риски, либо создают для себя максимально безопасные условия.

В настоящее время требования к безопасности со стороны потребителей достаточно высоки, и оптимальное решение состоит в полноценном использовании встроенных средств безопасности и разумным их дополнением продуктами и решениями сторонних разработчиков. Следствием этого являются слабое владение информацией о возможностях встроенных средств защиты Access Control и других систем и их корректного использования. Другим следствием является та же ситуация, но уже по отношению к продукции других производителей программно-аппаратных средств защиты информации и их использованию совместно с технологиями и продуктами Access Control Выход из такой ситуации, прежде всего, в подготовке кадров, обладающих экспертными знаниями в собственно информационной безопасности, в линейке продуктов Access и умеющих интегрировать разработки российских компаний со встроенными средствами защиты. Подобное обучение должно начинаться уже в профильных Вузах, и специалисты в данной области должны иметь возможность наращивать опыт и навыки в обучающих центрах. Хотелось бы видеть поддержку в данном вопросе как со стороны Access , так и со стороны других производителей, работающих на рынке информационной безопасности России.

В этой связи, очень обнадеживающей тенденцией, на наш взгляд, является, появление реальных решений, методов и подходов к организации

систем ИБ, разработанных отечественными компаниями совместно с российскими представительствами западных корпораций. Такое сотрудничество позволяет обеспечить не только устойчивую работоспособность механизмов защиты в составе информационной системы, но и соответствие этих решений требованиям российского законодательства.

Литература.

1. Hinz S., DuBois P., Stephens J., et al. MySQL 5.5 Reference Manual [Электронный ресурс]. Режим доступа: http://dev.mysql.eom/doc/refman/5.5/en/index.html

2. Haines R. The SELinux Notebook — The Foundations. 2nd Edition [Электронный ресурс]. Режим доступа: http://www.freeteehbooks.eom/efiles/selinuxnotebook/The_SELinux_ Notebook_Volume_1_The_Foundations.pdf

3. Smalley S. Configuring the SELinux Policy [Электронный ресурс]. Режим доступа: http: //www.nsa.gov/researeh/_files/selinux/papers/poliey2.pdf

4. Loscocco P. A., Smalley S. D. Meeting Critical Security Objectives with Security-Enhanced Linux [Электронный ресурс]. Режим доступа: http://www.nsa.gov/researeh/_files/ selinux/papers/ottawa01.pdf

УДК 004.94