А.С. Гуничева, Д.С. Гуменчук

ОБЕСПЕЧЕНИЕ ЗАЩИТЫ БАЗ ДАННЫХ С ПОМОЩЬЮ ПРИВИЛЕГИЙ И РОЛЕЙ

(Самарский университет)

В загруженных информационных системах, как наше общество, сдерживать количество данных почти нереально, что способствует деструктивному воздействию на них. Введем понятие «баз данных» в нашу статью и проведём аналогию риска безопасности информационных систем с риском безопасности «баз данных». Решить проблему доступа к базам данных способствовало введение таких конструкций как «роль» и «привилегии». В данной статье будет описан смысл установления пользователям ролей и привилегий в MySQL и приведены основные типы атак, связанных с данными понятиями. MySQL подходит для реализации решения больших и малых приложений.

Под угрозой безопасности информации понимается действие или событие, которое может привести к уничтожению, искажению или несанкционированному обороту информации. По происхождению источника угроз можно выделить четыре типа:

1) Угрозы, вызванные человеческим фактором..

2) Угрозы, источником которых являются внутренние программно – аппаратные средства.

3) Угрозы, источником которых являются несанкционированно встроенные программно-аппаратные средства.

4) Угрозы, обусловленные средой обитания .

Специфичными для систем управления базами данных угрозами доступности являются:

- Использование свойств первичных и внешних ключей;
- Блокировка записей при изменении;
- Атаки на переполнение буфера;
- Использование вредоносных программ;
- Загрузка системы пустой работой

Дискреционная модель реализуется с помощью базовых для нее понятий: привилегия и роль. Привилегия – это разрешенное действий или право доступа к определенным объектам СУБД. Привилегии делятся на системные и объектные. Рекомендовано не давать обычным пользователям использовать привилегию ANY (например, UPDATE ANY TABLE). Объектные привилегии, в свою очередь, связаны с различными типами объектов. Объединением нескольких привилегий в поименованный набор являются роли, которые обычно создаются администратором и выдаются другим пользователям. Для того, чтобы выдавать роль, на это также необходимо иметь соответствующую привилегию CREATE ROLE. Перед тем, как пользователь сможет воспользо-

ваться ролью, ее нужно включить для него. Сразу после создания роли нет никаких привилегий. Синтаксис создания роли Oracle/PLSQL:

CREATE ROLE role_name [NOT IDENTIFIED | IDENTIFIED {BY password | USING [schema.] package | EXTERNALLY | GLOBALLY};

где role_name - наименование новой роли, которую вы создаете.

NOT IDENTIFIED - роль немедленно включена. Не требуется ни один пароль, чтобы включить роль. IDENTIFIED - пользователь должен быть авторизован, прежде чем роль будет включена. BY password - пользователь должен ввести пароль, чтобы включить роль. USING package – вы создаете роль приложения — роль, которая включена только в приложениях с использованием авторизованного пакета. EXTERNALLY - пользователь должен быть авторизован внешним сервисом для включения роли. GLOBALLY - пользователь должен быть авторизован службой каталогов предприятия.

Синтаксис для предоставления привилегий таблице на роль в Oracle/PLSQL:

GRANT privileges ON object TO role_name;

где object - наименование объекта базы данных, которому вы предоставляете привилегии.

role_name - название той роли, которой будут предоставлены эти привилегии.

Для отмены любой из привилегий, вы можете выполнить команду revoke. Синтаксис для отмены привилегий таблицы на роль в Oracle/PLSQL:

REVOKE privileges ON object FROM role_name;

Синтаксис предоставления роли пользователю в Oracle:

GRANT role_name TO user_name;

role_name - название роли, которую вы хотите предоставить.

user_name - имя пользователя, которому будет предоставлена роль.

Чтобы включить или отключить определенную роль для текущей сессии, вы можете использовать оператор SET ROLE.

Когда пользователь входит в Oracle, все роли по умолчанию включены, но роли не по умолчанию должны быть включены с помощью оператора SET ROLE. Синтаксис для оператора SET ROLE в Oracle:

SET ROLE (role_name [IDENTIFIED BY password] | ALL
[EXCEPT role1, role2, …] | NONE);

где role_name - название роли, которую вы хотите включить.

IDENTIFIED BY password - пароль для роли, чтобы ее включить. Если роль не имеет пароля, этот параметр может быть опущен.

ALL - все роли должны быть включены для этой текущей сессии, за исключением тех, которые перечислены в EXCEPT.

NONE - отключает все роли для текущей сессии (включая все роли по умолчанию).

Oracle содержит несколько стандартных ролей, такие как CONNECT, RESOURSE и DBA, которые призваны помогать администратору в управлении базами данных. Особое внимание стоит обратить на роль DBA, которая содержит в себе все системные привилегии с параметрами ADMIN OPTION.

Также существуют такие роли как: EXP_FULL_DATABASE, которая предоставляет привилегии, требуемые для выполнения полного и инкрементного экспорта базы данных; IMP_FULL_DATABASE – предоставляет привилегии, требуемые для выполнения полного импорта базы данных; DELETE_CATALOG_ROLE – предоставляет привилегию DELETE на таблицу системного аудита; ECXECUTE_CATALOG_ROLE – предоставляет привилегию EXECUTE на объекты в словаре данных + роль HS_ADMIN_ROLE; SELECT_CATALOG_ROLE – предоставляет привилегию SELECT на объекты в словаре данных роль + HS_ADMIN_ROLE; RECOVERY_CATALOG_ROLE – предоставляет привилегии владельца каталога восстановления; HS_ADMIN_ROLE – используется для защиты доступа к таблицам и пакетам словаря данных. При всем многообразии стандартных ролей, для обеспечения безопасности рекомендуется создавать свои собственные роли, не особо полагаясь на встроенные.

При входе пользователя в систему Oracle включает все привилегии, явно предоставленные пользователю и все привилегии в ролях пользователя по умолчанию. Список ролей пользователя по умолчанию можно устанавливать и изменять оператором ALTER USER. Этот оператор позволяет указывать роли, которые должны быть включены автоматически, не запрашивая пароль. Причем эти роли должны быть уже непосредственно предоставлены оператором GRANT. По умолчанию нельзя устанавливать внешние и глобальные роли. Важный момент: когда вы создаете роль (не роль пользователя), она присваивается вам неявно, то есть добавляется как роль по умолчанию. Здесь может появиться коварная ошибка при попытке соединения с базой данных. Связана она с таким параметром как MAX_ENABLED_ROLES, который отвечает за максимальное количество ролей, которые может включить пользователь. Ошибка возникает при превышениколичества включенных ролей для пользователя. Ее можно избежать, изменив количество ролей по умолчанию так, чтобы их число было меньше данного параметра.

Синтаксис для установки роли по умолчанию в Oracle:

```
ALTER USER user_name DEFAULT ROLE ( role_name | ALL
[EXCEPT role1, role2, …] | NONE);
```

где user_name - имя пользователя, роль которого вы устанавливаете, как DEFAULT.

role_name - название той роли, которую вы хотите установить, как DEFAULT.

ALL - все роли должны быть как DEFAULT, за исключением тех, которые перечислены в EXCEPT.

NONE - запрещает все роли DEFAULT.

Синтаксис удаления роли в Oracle:

```
DROP ROLE role_name;
```

где role_name - название роли, которая должна быть удалена.

На основе вышеизложенных фактов можно сделать вывод о том, что роли и привилегии являются мощным инструментом защиты баз данных. При-

чем, наибольшая эффективность достигается при их комплексной и продуманной реализации.

Литература

1. Смирнов, С.Н. Безопасность систем баз данных/С.Н.Смирнов .— М.: Гелиос АРВ, 2007, — 352 с.

Е.Э. Елисеев

АДАПТИВНЫЙ АЛГОРИТМ В СИСТЕМЕ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ SNORT ДЛЯ ПРЕДОТВРАЩЕНИЯ WEB УГРОЗ

(Самарский университет)

Система обнаружения вторжений (сокращённо СОВ) – программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими через Интернет. СОВ обеспечивают дополнительный уровень защиты компьютерных систем [2].

СОВ используются для обнаружения некоторых типов вредоносной активности, которая может нарушить безопасность компьютерной системы. К такой активности, в том числе, относятся:

1. Сетевые атаки против уязвимых сервисов.
2. Атаки, направленные на повышение привилегий.
3. Неавторизованный доступ к важным файлам.
4. Действия вредоносного программного обеспечения (компьютерных вирусов, троянов и червей).

Обычно архитектура СОВ включает в себя четыре основных подсистемы:

1. Сенсорная подсистема, предназначенная для сбора событий, связанных с безопасностью защищаемой системы.
2. Подсистема анализа, предназначенная для выявления атак и подозрительных действий на основе данных сенсоров.
3. Хранилище, обеспечивающее накопление первичных событий и результатов анализа.
4. Консоль управления, позволяющая конфигурировать СОВ, наблюдать за состоянием защищаемой системы и СОВ, просматривать выявленные подсистемой анализа инциденты.

Каждая подсистема непосредственно связана с остальными, работа СОВ заключается в их совместном функционировании.

Существует несколько способов классификации СОВ в зависимости от типа и расположения сенсоров, а также методов, используемых подсистемой анализа для выявления подозрительной активности.