Слайд 1

Системы управления базами данных являются необходимыми программными продуктами при построении информационных систем. Однако в этих системах СУБД часто являются основными целями атак злоумышленников.

Для предотвращения нежелательного доступа к информации в базах данных используются различные подходы к реализации безопасности. Одним из подходов является аутентификация — процедура проверки подлинности. Правильно выбранный метод аутентификации в СУБД позволяет минимизировать шанс взлома, а следовательно, и предотвратить предотвратить несанкционированный доступ к информации, хранящейся в ней.

Одной из наиболее развитых систем управления базами данных является PostgreSQL. PostgreSQL — это свободно распространяемая объектно-реляционная система управления базами данных. Благодаря открытой лицензии и подробной документации PostgreSQL широко используется в различных информационных системах.

В учебно-исследовательской работе предлагается обзор основных методов аутентификации в системе управления базами данных PostgreSQL, производится выбор метода аутентификации пользователей СУБД в распределенной многопользовательской системе с использованием сервера OpenLDAP и описывается процесс настройки данного метода аутентификации. Целью данной учебно-исследовательской работы является исследование принципов работы сервера OpenLDAP и возможностей дальнейшей его доработки для интеграции с системой принудительного контроля доступа SELinux.

Слайд 2

В данной работе были исследованы основные методы аутентификации СУБД PostgreSQL:

- Trust
- Password
- Ident
- Peer
- PAM
- LDAP

Метод **Trust** позволяет любому, кто подключится к серверу баз данных получить доступ к любой базе данных, включая базу данных администратора. *Достоинства*: не требуется создание дополнительных пользователей ОС. Соответственно, в качестве основного *недостатка* является отсутствие возможности разграничить доступ пользователей к СУБД.

Группа методов **Password** включает в себя методы md5, crypt и password. В данных методах каждому пользователю при создании создается пароль, который хранится к системной таблице pg_shadow. При подключении к серверу СУБД пользователь вводит пароль от своего аккаунта, после чего СУБД выполняет поиск на соответствие пользователя введенному паролю в данной таблице. *Достоинства*: не требуется создание дополнительных пользователей ОС. Основным *недостатком* данных методов является передача пароля по нешифрованному каналу.

Работа метода аутентификации **Ident** следующая: при подключении пользователя PostgreSQL сверяет имя пользователя с именем пользователя СУБД. Если совпадение пользователей ОС и СУБД успешно, то доступ разрешается. *Достоинства*: используется в архитектуре клиент-сервер, простота архитектуры. *Недостатки*: данный протокол не предназначен для аутентификации или контроля доступа, а также существует вероятность подмены сервера/хоста.

Метод **Peer** работает следующим образом: имя пользователя ОС сопоставляется с именем базы данных. *Достоинства*: простота архитектуры. *Недостатки*: применяется исключительно для локальных соединений.

PAM (Pluggable Authentication Modules — «подключаемые модули аутентификации») представляют собой набор разделяемых библиотек, которые позволяют интегрировать различные низкоуровневые методы аутентификации в виде единого высокоуровневого API.

Приложениям, в т.ч. и PostgreSQL, которым необходимо выполнить аутентификацию пользователя обращаются в службу PAM. PAM проводит процедуру аутентификации и возвращает результат: PAM_SUCCESS или PAM_AUTH_ERR. PAM состоит из динамических библиотек (они проводят процедуру аутентификации) и конфигурационных файлов (в них описан порядок использования модулей при аутентификации). *Достоинствами* метода является простота реализации, расширяемость, поддерживает клиент-серверную архитектуру. Основным *недостатком* является подмена модуля и необходимость тонкой настройки.

Метод **LDAP**. Данный метод аутентификации работает аналогично методам аутентификации по паролю, за исключением того, что используется LDAP для проверки подлинности. При этом серверу LDAP высылается имя пользователя и его пароль, после чего он выполняет поиск записи по этим данным. Если такая запись найдена, то пользователю разрешается доступ в базу данных, иначе — запрещается. *Достоинствами* являются: централизованное хранилище аккаунтов пользователей, что не требует создания локальных пользователей ОС, поддержка клиент-серверной архитектуры, расширяемость, шифрование соединения по протоколу TLS. В качестве *недостатков* можно отметить необходимость постоянно работающего сервера LDAP для аутентификации пользователей.

Слайд 3

LDAP - это аббревиатура от *Lightweight Directory Access Protocol*. Как следует из названия, это облегченный протокол доступа к службам каталогов, предназначенный для доступа к службам каталогов. LDAP работает поверх TCP/IP или других ориентированных на соединение сетевых протоколов.

Каталоги LDAP используют модель данных, которая считает или представляет данные как иерархию объектов. Это не означает, что LDAP является объектно-ориентированной базой данных.

В LDAP-каталоге данные представлены как иерархия записей. Полученная в результате древовидная структура называется **информационным деревом каталога** (*Data Information Tree*, *DIT*). Верхнюю часть данного дерева обычно называют **корнем** (**root**), (а также базой (**base**) или суффиксом (**suffix**)).

Каждый элемент DIT называется записью (**object**). Каждая запись имеет ноль или более дочерних записей. Каждая дочерняя запись является одноуровневой (братской) по отношению к другим дочерним записям своей родительской записи. Каждая запись является экземпляром одного или нескольких объектных классов (**objectClass**).

Объектные классы содержат ноль или более атрибутов (**attribute**). Атрибуты имеют имена (и, иногда, аббревиатуры или псевдонимы) и обычно содержат данные.

OpenLDAP — открытая реализация протокола LDAP.

`slapd` — сервер службы каталогов, реализующую 3ю версию протокола LDAP. В качестве хранилища поддерживаются механизмы манипуляции данных. Одним из самых распространенных является BDB (высокопроизводительный механизм манипуляции с поддержкой транзакций) на базе Berklery DB. Данные в этой БД хранятся в виде ключ-значение.

Слайд 4

В рамках учебно-исследовательской работы была поставлена задача развернуть стенд из трех машин, демонстрирующий принцип работы метода аутентификации LDAP в PostgreSQL.

Стенд представляет собой 3 машины: клиента, сервера LDAP, сервера СУБД PostgreSQL.

Условно работу стенда можно разделить на 2 процесса: аутентификация пользователя на клиентской машине и аутентификацию пользователей в СУБД.

Принцип работы стенда следующий:

1. Пользователь вводит имя своего аккаунта на клиентской машине и пароль. Пароль отсылается LDAP-серверу в закрытом виде и проверяется на совпадение.
2. Если переданный и хранимый пароль совпадают, пользователь успешно проходит аутентификацию на клиенте.
3. При подключении к базе данных с помощью клиента `psql`, пользователь передает свой `uid` (логин пользователя) и пароль в PostgreSQL.
4. PostgreSQL генерирует запрос на основе данных клиента.
5. LDAP-сервер выполняет поиск записи и сверяет пароль. Результат поиска отправляется PostgreSQL.
6. Если такой записи нет или пароль неверный, в подключении к базе данных отказывается.

Соединение между машинами шифруется с помощью SSL/TLS.

Слайд 5

Каждому пользователю создаются записи на сервере LDAP. На слайде приведена запись пользователя Nikolay Ivanov. `dn` (*Distrbution name*) — путь к записи в каталоге, `cn` (*Common Name*) — имя пользователя. С помощью директив `objectClass` указывается принадлежность атрибутов записи к объектным классам, `shadowLastChange`, `shadowMin`, `shadowMax`, `shadowWarning` — служебные поля.

Слайд 6

Настройка аутентификации пользователей на клиентской машине реализуется с помощью демона **SSSD** (*System Security Services Deamon* — демон сервисов системной безопасности). Он позволяет обращаться к удаленным механизмам аутентификации.

На слайде приведен вывод команды `getent`. Она позволяет получить информацию о конкретном пользователе.

Слайд 7

Шифрование соединения достигается с помощью TLS-соединения. Для этого создаются SSL-ключи и их пути заносятся в каталог LDAP. Дополнительно опцией демона SSSD указывается активность соединения с LDAP с использованием шифрования. Слева приведен вывод утилиты `tethreal` (показывает сетевую активность интерфейса и порта) при использовании нешифрованном соединении и с с использованием шифрования по протоколу TLS (*Transport Socket Layer* — «уровень защищенных сокетов»).

Слайд 8

Файл `pg_hba.conf` представляет собой таблицу с правилами аутентификации пользователей. Для установки аутентификации `ldap` используется следующая строка.

Она означает, что разрешено подключение ко всем базам данных всем пользователям, IP-адреса клиентов которых находятся в подсети `192.168.100.0` при успешной аутентификации по методу `ldap`. При этом подключение между клиентом и сервером будет зашифровано по протоколу SSL (на это указывает параметр `hostssl`).

В качестве аргументов метода аутентификации `ldap` указывается IP-адрес LDAP-сервера — `192.168.100.3`, префикс `ldapprefix` — идентификатор записи в каталоге LDAP, а также суффикс `ldapsuffix` — адрес записи в каталоге.

Слайд 9

Данная учебно-исследовательская работа посвящена исследованию возможностей сервера OpenLDAP для аутентификации пользователей баз данных PostgreSQL.

Были достигнуты следующие основные результаты:

1. Произведено исследование основных методов аутентификации в PostgreSQL. Выявлены их достоинства и недостатки.
2. Исследованы возможности сервера OpenLDAP.
3. Описан процесс настройки метода аутентификации LDAP в PostgreSQL.

Таким образом, можно судить о достижении поставленной цели. Продолжением темы учебно-исследовательской работы может послужить программная реализация механизма хранения метки SELinux в OpenLDAP и ее передачи для присвоения контекста безопасности пользователям операционных систем и баз данных.