\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ  Кафедра №42  «Криптология и кибербезопасность» |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Федеральное государственное автономное образовательное учреждение высшего образования*

**«Национальный исследовательский ядерный университет «МИФИ»»**

ЛАБОРАТОРНАЯ РАБОТА №2-4:

**«Аутентификация и базовый контроль доступа.»**

Аверин Владислав

Группа: Б19-505

Ноябрь, 2022

*Содержание*

[*1. SCHEMA-ONLY пользователь* 4](#_Toc120875835)

[*2. Аутентификация средствами ОС* 7](#_Toc120875836)

[*3. Аутентификация посредством радиус-сервера* 10](#_Toc120875837)

[*4. Использование представлений для разграничения доступа* 14](#_Toc120875838)

[*Выводы:* 16](#_Toc120875839)

*Цель работы*

Рассмотреть различные подходы к аутентификации пользователей. Рассмотреть примитивные механизмы разграничения доступа к столбцам и записям базы данных.

*Ход работы*

1. Создать schema-only пользователя, добавить один или несколько объектов в его схему данных, убедиться в их успешном создании и работоспособности. Обосновать, почему (не) планируется использовать пользователя такого типа для хранения объектов базы данных в разрабатываемой схеме;
2. Создать пользователя, проходящего аутентификацию средствами операционной системы. Важно: имя пользователя должно начинаться с OS AUTHENT PREFIX (по умолчанию ‘OPS$’). Если работа выполняется с учётной записи администратора, необходимо создать ещё одну учётную запись в операционной системе, так как административная учётная запись уже связана с SYSDBA;
3. (дополнительное кармическое задание) Развернуть RADIUS-сервер. Создать пользователя, который проходит аутентификацию посредством RADIUS-сервера. Важно: настройки сервера СУБД для работы с RADIUS-сервером могут конфликтовать с настройками для аутентификации средствами операционной системы. Необходимо убедиться, что возможно войти в учётную запись администратора без использования этого механизма;
4. Ограничить доступ пользователей/ролей к особо важным данным за счёт использования представлений. Убедиться в работоспособности решения;
5. Оформить отчёт.

# *SCHEMA-ONLY пользователь*

Пользователь данной категории хоть и может иметь привилегии на подключение, но, по сути, не может подключаться к БД, т.к. для него не настроено ни одного способа аутентификации.

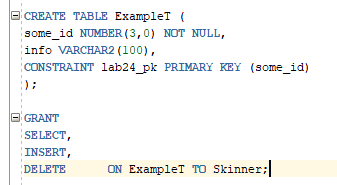
Порядок действий (не уверен, что верный, но вроде работает):

1. Создание обычного пользователя и выдача ему необходимых для администрирования собственных таблиц привилегий:





Подключение к нему и создание от его лица таблицы, с выдачей необходимых привилегий другому пользователю:

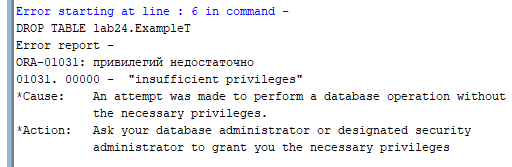


Как видно, привилегий ALTER и DROP TABLE у схемы Skinner нет, поэтому сторонний пользователь сможет только модифицировать данные таблицы, но не саму таблицу:









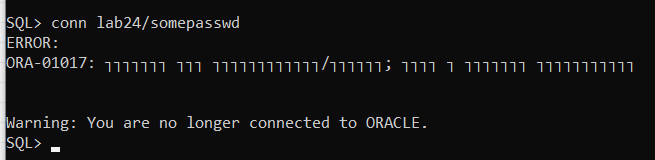
Меняем от лица sysdba режим







Попытки подключиться приведут к ошибке пароля: способа аутентификации (и самого пароля) то нет



(Note: насколько я понял, к нему можно подключаться через CONNECT THROUGH: но опять же, для этого нужны соответствующие привилегии, которых ни у кого, кроме sysdba по идее, быть не должно.)

Механизм schema-only пользователя, насколько я понимаю, введен для обеспечения безопасности. Если каждая схема пользователя может владеть совокупностью объектов, а каждый объект должен принадлежать какой-либо схеме (а-ля один ко многим), то при несанкционированном доступе к данной схеме (даже простым брутом пароля, если администратор неправильно настроил СУБД и IDS) злоумышленник может просто удалить таблицу и возможно ее бэкапы (хотя, удалить все данные из самой таблицы все еще никто не мешает). Централизированное же управление всеми объектами сужает вектор атаки: возможность поломки данных и самой БД появляется только при взломе данного пользователя или sysdba. И если в случае с sysbda уже класть на всю защиту, это фактически полный доступ ко всему, то исключение подключения к владеющей таблицами схеме уменьшает этот вектор до одного администратора. А взломать schema-only пользователя нельзя: в него технически нельзя зайти, пока это не изменит системный администратор. Правда, непонятно, почему все таблицы в таком случае не создавать от имени sysdba, если все равно его взлом означает полный контроль над БД (можно просто изменить AUTHENTI COM для schema-only пользователя и подключиться к нему). Возможно, дело в квоте памяти и производительности, а также для удобства, когда таблиц слишком много и они загромождают пространство главного администратора. Поэтому если предположить, например, что администратор управляет сразу несколькими БД, то для сегментации памяти каждой базы данных логичнее было бы создать своих schema-only пользователей.

Более того, обеспечение частичного доступа к объектам для всех пользователей системы защищает кроме как атак извне, и от атак изнутри. Если какой-либо инсайдер из числа доверенных лиц (если не брать самого системного администратора) окажется очень “хорошим” (или очень криворуким) человеком, то навредить самим объектам настолько критично (будь то хоть бекапы или индексы с различными ролями) он не сможет.

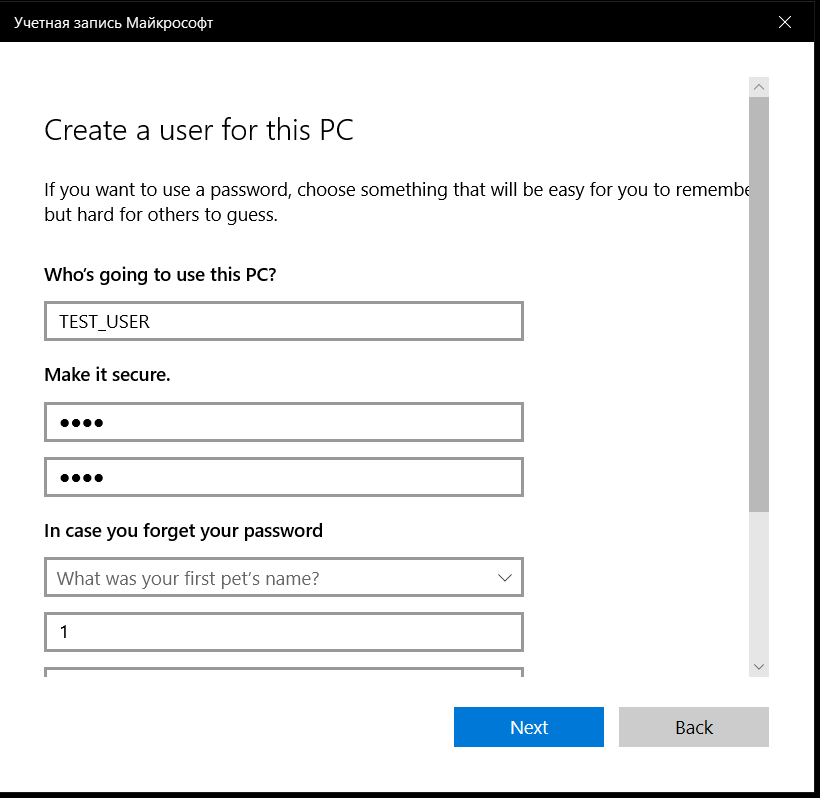
Механизм включения привилегий входа в систему, насколько я знаю, организованы во многих других продуктах, будь то разные ОС, другие СУБД или сервера. Но за достаточно простой структурой БД использование таких пользователей в наших лабах не видится рентабельным (я и с наличием schema-only пользователя все равно справлюсь с поломкой СУБД :))

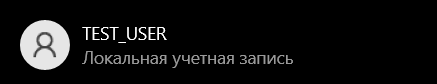
# *Аутентификация средствами ОС*

Информация взята из: <https://oracle-base.com/articles/misc/os-authentication>

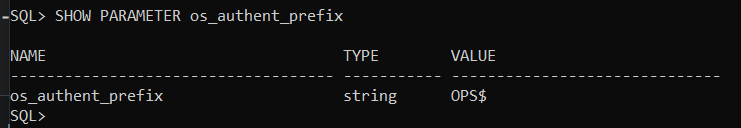
Для реализации доступа средствами операционной системы, создадим нового пользователя в Windows:

ВАЖНО! Лично у меня не получалось подключиться к пользователю, если его имя было указано в lowercase. Скорее всего это из-за того, что внутри БД пользователи хранятся в верхнем регистре.

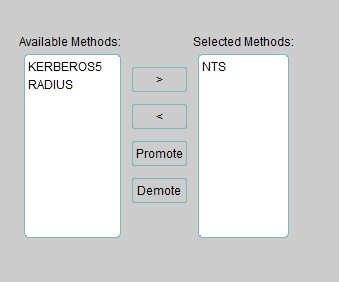




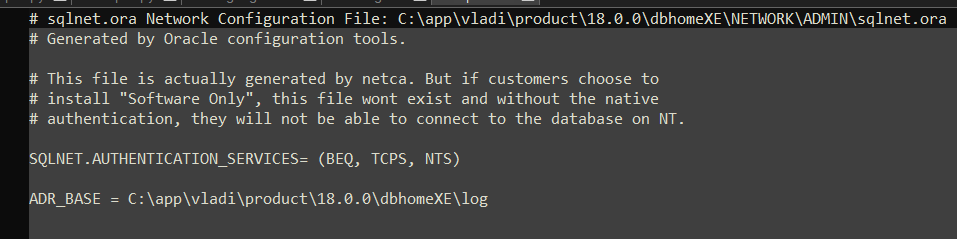
Проверим значение параметра инициализации для аутентификации средствами ОС:



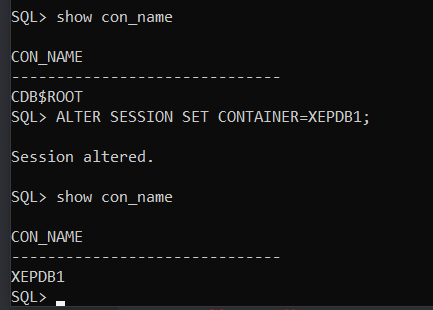
Проверим х2, что способ аутентификации NTS (раз мы используем Windows) активен:



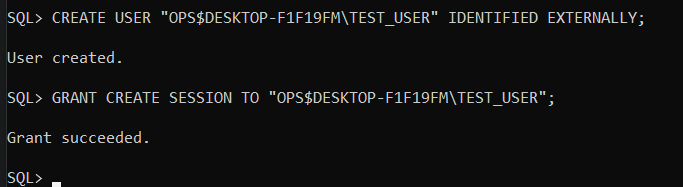
Ну или прописать напрямую в файле:



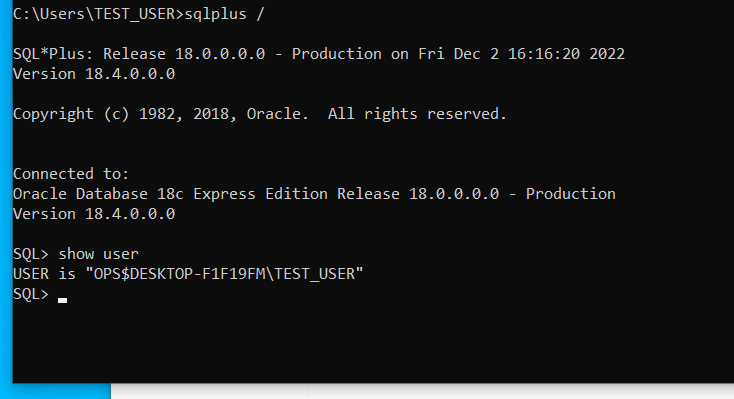
Проверим х3, что выбран контейнер XEPDB1, т.к. при использовании NTS (вход по “sqlplus / as sysdba” он автоматом перекидывает в корневой контейнер):



И только теперь можем создавать пользователя с именем: “OPS$DOMAIN-NAME\TEST\_USER”; вместо DOMAIN\_NAME в случае, если компьютер (пользователь) находится не в домене, указывается имя компьютера:



Зайдем из-под имени TEST\_USER и проверим, что оно таки работает:



# *3. Аутентификация посредством радиус-сервера*

Изначально меня почему-то переклинило на то, что в задании было написано про IDENTIFIED GLOBALLY (т.е. про вход через Windows AD). А так как мы уже делали в этом семестре по одной из дисциплин, связанных с разворачиванием домена в Microsoft Active Directory, то я начал пытаться развернуть на том windows server радиус сервер, и добавить СУБД туда. Спойлер: ниче не вышло :-) Ибо в лабораторной у нас была только самая тривиальная последовательная настройка домена для мартышек (чтобы познакомиться с СЗИ Secret Net Studio). Поэтому не мудрствуя лукаво я просто взял свою старую гостевую Убунту и конечно же сам*,* без помощи всяких запрещенных BitTorrent порталов, которыми пользоваться категорически нельзя и вообще, пиратство это плохо (тем более, что по факту, никаких прав нарушено не было), поднял гостевую ОС Oracle Linux 8.5 с установленной и настроенной Oracle EX 21c. Обе ОС (и убунту, и Оракл) запускались через гипервизор Virtual Box с настройками сетевого моста (просто чтобы они могли видеть друга). В качестве радиус сервера был выбрал FreeRADIUS. Собственно, информация для установки и конфигурировании серверов оракла и freeradius была взята из документации и еще пары источников ниже (Спойлер х2: из задания ниче не вышло х2, так что можно 3ий пункт пропускать :-)):

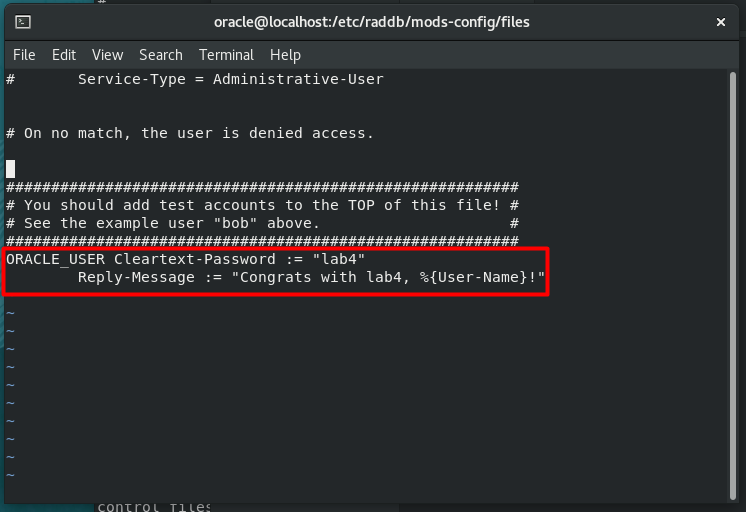
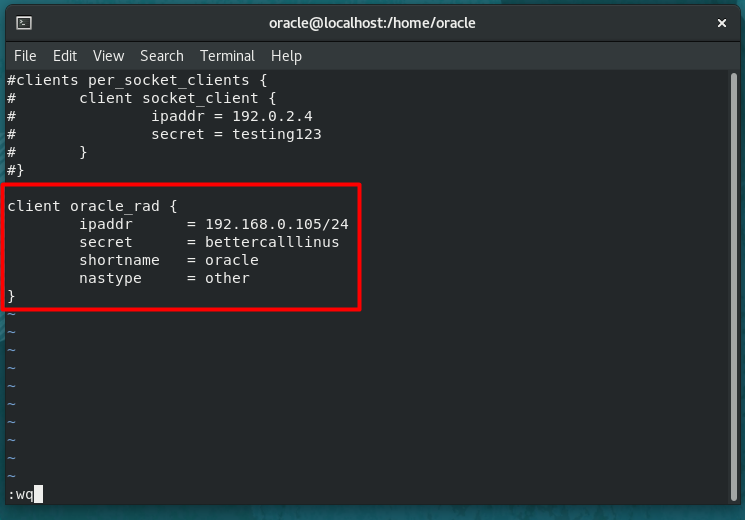
<https://docs.oracle.com/database/121/DBSEG/asoradus.htm#DBSEG9633>

<https://www.dmosk.ru/miniinstruktions.php?mini=freeradius-centos8>

<https://blog.pythian.com/using-freeradius-to-authorize-oracle-connections/>

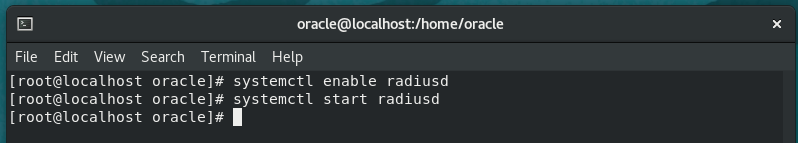
1. Настройка радиус-сервера.

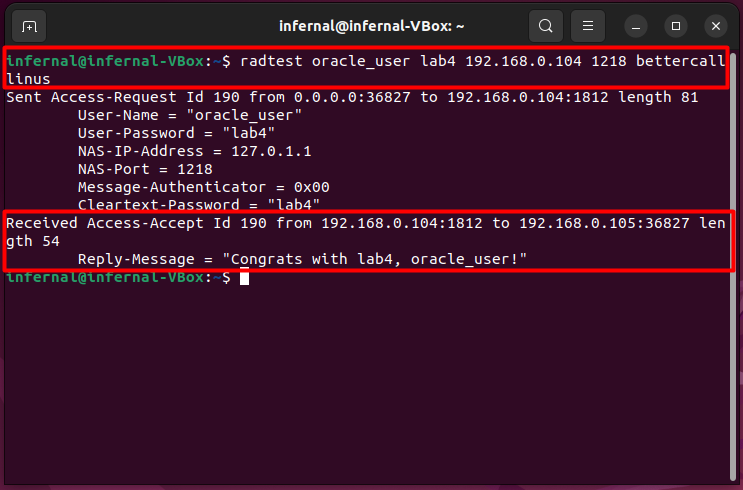
Для начала нужно было установить на обе машины freeradius (на oracle linux еще freeradius-utils, хотя может и не надо было). После добавим какого-нибудь пользователя в /etc/raddb/users и новый хост (нашу гостевую убунту) в /etc/raddb/clients.conf (в большинстве rpm ориентированных сборок freeradius лежит в /etc/freeradius/’номер версии’/, но в оракле почему-то была просто папка etc/raddb):

Имя пользователя для оракла обязательно должно быть в upper case (по аналогии с пользователем Windows), т.к. freeradius различает регистры в отличие от oracle.

Запустим демона radiusd и попробуем сделать запрос к радиус серверу с гостевой убунты (ip oracle linux 192.168.0.104, убунты – 105):

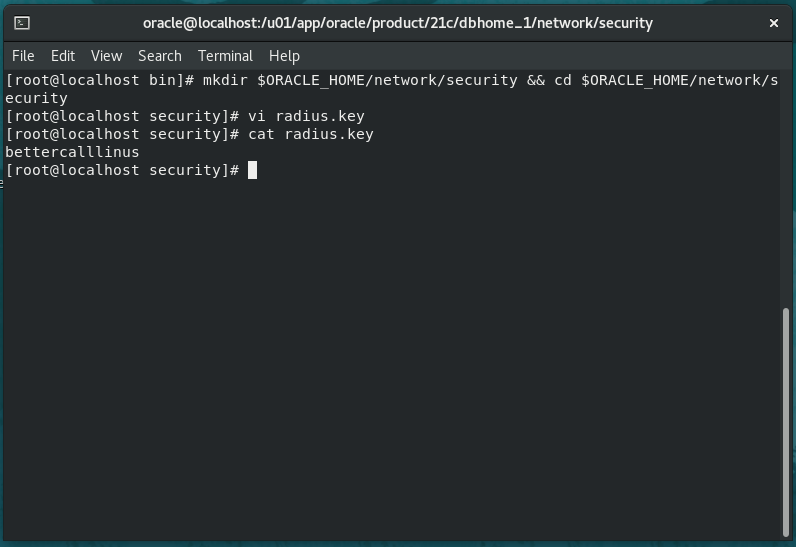


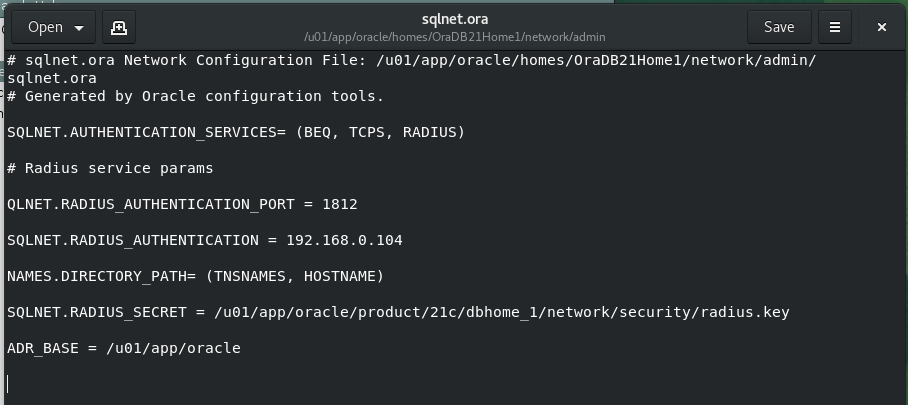


Первичная настройка радиус сервера завершена. Теперь перейдем к конфигурации сервера СУБД.

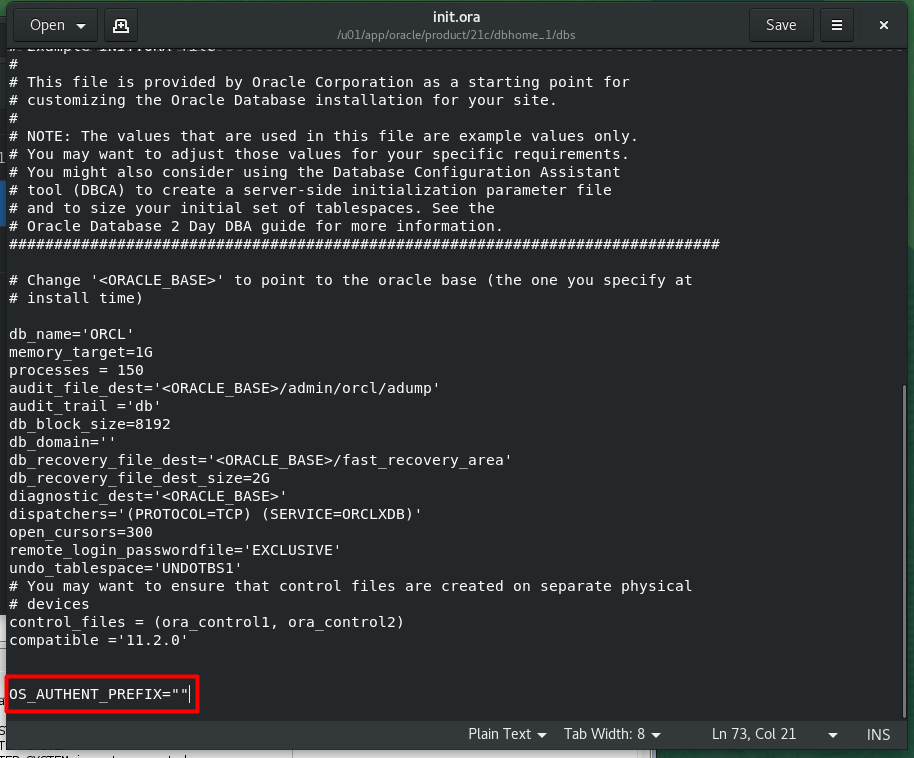
1. Конфигурация Радиус авторизации на сервере СУБД Oracle (да-да, дословный перевод названия 2ого пункта из документации оракла)

Собственно, делаем то, что написано в первых двух пунктах: добавляем в SQLNET.AUTHENTICATION\_SERVICES метод аутентификации RADIUS и radius.key секрет нашего созданного пользователя.



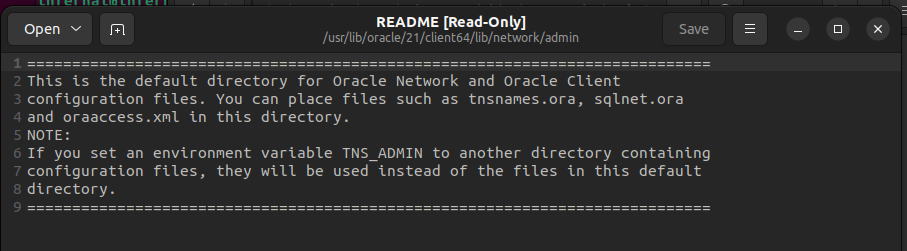


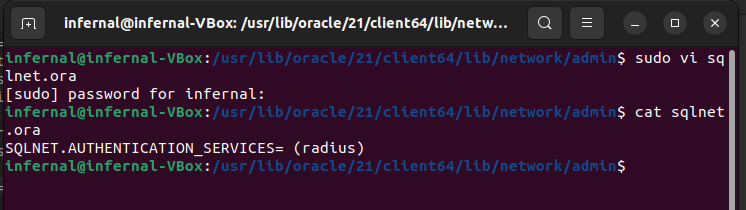
Добавляем параметр аутентификации:



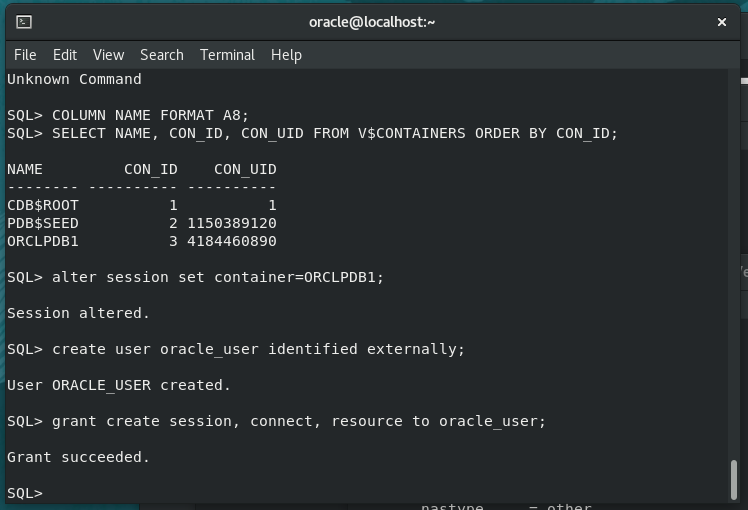
Устанавливаем Oracle Instant Client на Убутну:

https://www.foxinfotech.in/2019/03/how-to-install-sqlplus-in-linux-ubuntu.html

Единственное, что немного отличается от этого мануала, это то, что самого файла sqlnet.ora по дефолту не было (с другой стороны, откуда ему там взяться без установленного экземпляра БД):



Теперь создаем пользователя для авторизации с параметром IDENTIFIED EXTERNALLY:

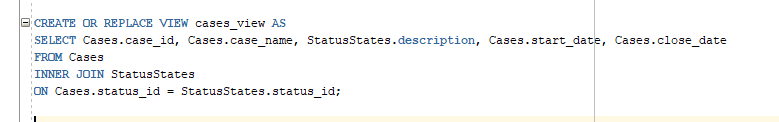


И…. Дальше я просто запутался (На самом деле, у меня возникли проблемы с самим freeradius, и я в течение написания этих лаб поломал свою убунту, так что я просто сдался)

Далее, по идее, нужно настроить freeradius на работу с oracle, поднять какой-нибудь веб-интерфейс (через апач или нджинкс), но скорее всего, не в этом семестре, простите :(

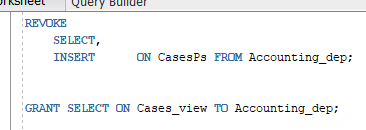
# *4. Использование представлений для разграничения доступа*

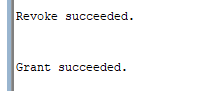
Создадим для примера представление, ограничивающее доступ к данным для пользователей, которые являются бухгалтерами. Например, им незачем знать подробности дел (будем считать, что им нужна только информация о статусе дела и связанных датах), поэтому атрибут уровня доступа и ссылки на дело можно скрыть от них:



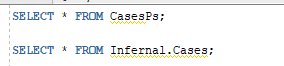


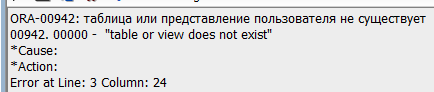
Изменим привилегии роли Accounting\_dep, забрав возможность работы напрямую с Cases, и дав право на использование созданного представления:





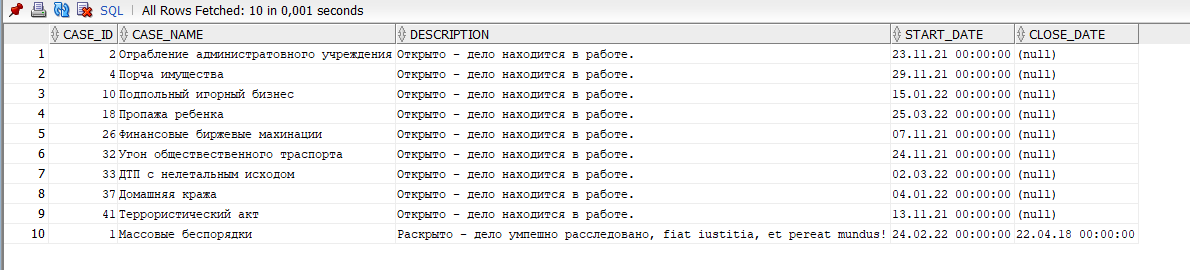
Переключимся на пользователя Nadya (которая с ролью accounting\_dep) и попробуем обратиться к таблице Cases:





А вот созданное представление она видит, и может задавать необходимые запросы:





# *Выводы:*

В результате данной лабораторной работы были изучены и протестированы различные способы аутентификации в СУБД Oracle, а также создано представление для разграничения доступа некоторых пользователей к не делегированным им данным. Также была предпринята попытка развертывания аутентификации внешними программными средствами (посредством RADIUS-сервера), и создан schema-only пользователь, который может использоваться для безопасной настройки владения ресурсами СУБД.