МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение

высшего образования

**«Северный (Арктический) федеральный университет имени М.В. Ломоносова»**

|  |
| --- |
| Высшая школа информационных технологий и автоматизированных систем |

**ЛАБОРАТОРНАЯ РАБОТА №2**

|  |  |  |
| --- | --- | --- |
| По дисциплине: | | Защита информации в системах управления базами данных |
|  | | |
|  | | |
| На тему | Системы хранения данных. Часть 2 | |
|  | | |
|  | | |

|  |  |
| --- | --- |
|  | Выполнил обучающийся:  Грозов Илья Владимирович |
|  |  |
|  | Направление подготовки / специальность:  10.03.01 Информационная безопасность |
|  |  |
|  | Курс: 3 |
|  | Группа: 151113 |
|  | Руководитель: Зубарев Александр Андреевич, ст. |
|  | преподаватель |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отметка о зачете |  |  |  |  |
|  |  |  |  |  |
| Руководитель |  |  |  | А.А. Зубарев. |
|  |  |  |  |  |

Архангельск 2023

##### Задание

Получить практический навык PostgreSQL CIS Benchmarks и ПРД

ХОД РАБОТЫ

**1. ПОДГОТОВКА КОНТЕЙНЕРА DOCKER-COMPOSE**

1.1 Остановка контейнера docker-compose.yml

Чтобы изменить метод аутентификации на изначальный требуется остановить контейнер docker при помощи команды docker-compose stop postgres. Остановка контейнера отображена на рисунке 1.1.1

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

Рисунок 1.1.1 – Остановка контейнера

1.2 Удаление контейнера docker-compose

Необходимо удалить контейнер docker-compose. Удаление контейнера docker-compose производится с помощью команды: sudo docker-compose rm. Удаление контейнера необходимо подтвердить. Удаление контейнера отображено на рисунке 1.2.1.

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 1.2.1 – Удаление контейнера

1.3 Возвращение исходной конфигурации docker-compose.yml

Вернем файл docker-compose.yml в изначальное состояние. Уберем метод аутентификации. Исходный файл docker-compose.yml с убранным методом аутентификации предоставлен на рисунке 1.3.1

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1.3.1 - Исходный файл docker-compose.yml

1.4 Запуск контейнера

Запустим контейнер docker по исходному файлу docker-compose.yml при помощи команды: docker-compose up. Запуск контейнера docker приведен на рисунке 1.4.1

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

Рисунок 1.4.1 – Запуск контейнера docker

Итог запуска контейнера docker приведен на рисунке 1.4.2.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1.4.2 – Итог запуска контейнера docker

**2. БАЗА ДАННЫХ SQL**

2.1 Подключение к базе данных

Выполним подключение к базе данных при помощи команды: docker-compose exec postgres psql -U app -d db. Подключение к базе данных отображено на рисунке 2.1.1.

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

Рисунок 2.1.1 - Подключение к базе данных

**3. ПОДГОТОВКА ДАННЫХ. РАБОТА С ОБОЛОЧКОЙ SQL**

3.1 Отображение существующих системных ролей базы данных

Для отображения существующих системных ролей воспользуемся командой: SELECT \* FROM pg\_roles. Отображение существующих системных ролей приведено на рисунке 3.1.1

Изображение выглядит как текст, снимок экрана, черный

Автоматически созданное описание

Рисунок 3.1.1 – Отображение системных ролей

3.2 Создание новой роли reader с правом входа

Создадим новую роль reader с правом входа при помощи команды: CREATE ROLE reader LOGIN PASSWORD 'secret'. Создание новой роли отображено на рисунке 3.2.1

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание

Рисунок 3.2.1 – Создание новой роли reader с правом входа

3.3 Проверка существующих баз данных

Проверим существующие бзаы данных в системе при помощи команды: SELECT \* FROM pg\_database. Существующие базы данных с их владельцами отображены на рисунке 3.3.1



Рисунок 3.3.1 – Существующие базы данных с их владельцами

3.4 Создание тестовой базы данных

Создадим тестовую базу данных при помощи команды: test CREATE DATABASE test. Создание тестовой базы данных отображено на рисунке 3.4.1.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Рисунок 3.4.1 – Создание тестовой базы данных

3.5 Подключение к созданной базе данных

Выполним подключение к тестовой базе данных при помощи команды: \c test. Подключение к тестовой базе данных отображено на рисунке 3.5.1.

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание

Рисунок 3.5.1 – Подключение к тестовой базе данных

3.6 Создание таблицы records в базе данных

Создадим таблицу records в базе данных при помощи команды: CREATE TABLE records (value TEXT, status TEXT, created TIMESTAMP DEFAULT CURRENT\_TIMESTAMP). Создание таблицы отображено на рисунке 3.6.1



Рисунок 3.6.1 – Создание таблицы records

3.7 Добавление записи в таблицу

В созданную таблицу records добавим запись содержания: INSERT INTO records(value, status) VALUES ('transfer money from 55\*\* \*\*\*\* 0001 to 42\*\* \*\*\*\* 0002', 'success'). Добавление записи отображено на рисунке 3.7.1



Рисунок 3.7.1 – Добавление записи в таблицу

3.8 Предоставление прав ролям базы данных на чтение содержимого

Предоставим права для роли reader на выполнение чтения содержимого таблицы records при помощи команды: GRANT SELECT ON records TO reader. Добавление прав на чтение таблицы отображено на рисунке 3.8.1



Рисунок 3.8.1 – Добавление прав на чтение таблицы

3.9 Просмотр предоставленных прав

Выполним просмотр предоставленных прав для роли при помощи команды: \dp. Предоставленные права отображены на рисунке 3.9.1

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Рисунок 3.9.1 – Просмотр предоставленных прав

**4. ПРОВЕРКА ПРД**

4.1 Получение оболочки для пользователя

Выполним получение оболочки psql для пользователя reader при помощи команды: docker-compose exec postgres psql -U reader -d test, предварительно произведя выход из базы данныз командой \q. Выход из базы данных предоставлен на рисунке 4.1.1. Получение оболочки для пользователя предоставлено на рисунке 4.1.2

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

Рисунок 4.1.1 – Выход из базы данных

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4.1.2 – Получение оболочки для пользователя

4.2 Выполнение запроса на чтение

Выполним запрос на чтение при помощи команды: docker-compose exec postgres psql -U app -d test -c «GRANT SELECT ON records TO reader;». Выполнение запроса на чтение предоставлено на рисунке 4.2.1

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4.2.1 – Выполнение запроса на чтение

4.3 Отмена прав ролей на чтение

Отменим права на чтение у роли reader при помощи команды: docker-compose exec postgres psql -U app -d test -c «REVOKE SELECT ON records FROM reader;». Отмена прав на чтение у роли reader предоставлена на рисунке 4.3.1.

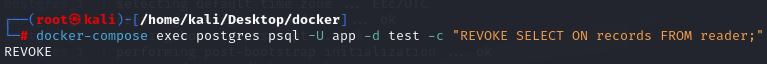


Рисунок 4.3.1 – Отмена прав на чтение

4.4 Выполнение запроса на чтение

Выполним запрос на чтение при помощи команды: SELECT \* FROM records, предварительно произведя обратный вход в базу данных. Ответ в запрос на чтение предоставлен на рисунке 4.4.1

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4.4.1 – Ответ в запрос на чтение

База данных сообщает нам об ошибке – отсутствуют права доступа. У пользователя от имени которого выполняется запрос отсутствуют права на доступ к таблице «records»

КОНТРОЛЬНЫЕ ВОПРОСЫ

**1. Какие методы не рекомендуется использовать для удалённых подключений?**

Для удаленных подключений к базе данных не рекомендуется использовать незащищенные методы, такие как открытые соединения или передача данных через незашифрованные протоколы, не рекомендуется использовать стандартные порты для удаленного доступа к базе данных

**2. Какие методы рекомендуется использовать для удалённых подключений?**

Для удаленных подключений к базе данных рекомендуется использовать защищенные методы, такие как VPN или SSH, следует удостовериться, что база данных настроена на безопасный режим работы с удаленными подключениями, например, путем ограничения доступа только к определенным IP-адресам

ВЫВОД

В ходе выполнения лабораторной работы по теме: «Работа со средствами дизассемблирования и откладки» получили практический навык PostgreSQL CIS Benchmarks и ПРД