Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет безопасности информационных технологий

Дисциплина:

«Информационная безопасность баз данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

«Шифрование в PostgreSQL»

Выполнили:

Чу Ван Доан, студент группы N3247

(подпись)

Проверил:

Волков А.Г.

(отметка о выполнении)

(подпись)

Санкт-Петербург

2024 г.

Содержание

1 Шифрование в PostgreSQL

1.1 Цель работы

Получение навыков шифрования в PostgreSQL.

1.2 Задание

1.2.1 Создайте таблицу, в которой два столбца содержат хешированные значения, где одно из них сгенерировано с помощью алгоритма SHA-1. Покажите, как можно выполнить проверку, используя данные двух хешей.

1.2.2 Создайте таблицу, в которой данные имеют байтовый тип. Зашифруйте этот столбец и покажите, как пользователь может расшифровать данные во время обычного select-запроса к зашифрованному столбцу.

1.3 Ход работы

1.3.1 Задание 1

Проверим список установленных расширений:

```
n3247_22_lab5=# \dx
                List of installed extensions
  Name   | Version |   Schema   |         Description
---------+---------+------------+------------------------------
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
(1 row)
```

Рисунок 1 – список установленных расширений

Устанавливаем расширение **pgcrypto:**

```
n3247_22_lab5=# create extension pgcrypto;
CREATE EXTENSION
n3247_22_lab5=# \dx
                 List of installed extensions
   Name   | Version |   Schema   |         Description
----------+---------+------------+------------------------------
 pgcrypto | 1.3     | public     | cryptographic functions
 plpgsql  | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
```

Рисунок 2 – Установка расширение **pgcrypto**

Создадим выделенную таблицу для хешированных значений и вставим в нее хеш с помощью следующих нескольких запросов:

```
n3247_22_lab5=# create table hash_password(id bigserial, value_hash1 text, value_hash2 text);
CREATE TABLE
n3247_22_lab5=# insert into hash_password (value_hash1, value_hash2) values
n3247_22_lab5-# (crypt('password1', gen_salt('md5')), digest('password2', 'sha1'));
INSERT 0 1
n3247_22_lab5=# select * from hash_password;
 id |            value_hash1             |               value_hash2
----+------------------------------------+------------------------------------------
  1 | $1$Ma08/JR9$Ec5vHKznoGIx90W9fpqEw. | \x2aa60a8ff7fcd473d321e0146afd9e26df395147
(1 row)
```

Рисунок 3 – Создание таблицы и вставка в нее хешированных значений

Проверим запросом введенный пароль.

```
n3247_22_lab5=# select (value_hash1 = crypt('mypassword', value_hash1)) as natch1,
(value_hash2::bytea = (digest('mypassword2', 'sha1'))) as match2 from hash_password;
 natch1 | match2
--------+--------
 f      | f
(1 row)

n3247_22_lab5=# select (value_hash1 = crypt('password1', value_hash1)) as match1,
(value_hash2::bytea = (digest('passwor2', 'sha1'))) as match2 from hash_password;
 match1 | match2
--------+--------
 t      | f
(1 row)

n3247_22_lab5=# select (value_hash1 = crypt('password1', value_hash1)) as match1,
(value_hash2::bytea = (digest('password2', 'sha1'))) as match2 from hash_password;
 match1 | match2
--------+--------
 t      | t
(1 row)
```

Рисунок 4 – Проверка введенных паролей

1.3.2 Задание 2

Создадим таблицу, в которой данные имеют байтовый тип.

```
n3247_22_lab5=# create table lab5_2 (id bigserial, data bytea);
CREATE TABLE
n3247_22_lab5=# insert into lab5_2 (data) values (pgp_sym_encrypt('Welcome to back', 'key1'));
INSERT 0 1
n3247_22_lab5=# select id, left(data::text, 30) from lab5_2;
 id |              left
----+--------------------------------
  1 | \xc30d040703023618981c098e1cf4
(1 row)
```

Рисунок 5 – Создание таблицы

Расшифруем данные ключом, который использовался при шифровании.

```
n3247_22_lab5=# select id, pgp_sym_decrypt(data::bytea, 'key1') as data from lab5_2;
 id |      data
----+-----------------
  1 | Welcome to back
(1 row)

n3247_22_lab5=# select id, pgp_sym_decrypt(data::bytea, 'key2') as data from lab5_2;
ERROR:  Wrong key or corrupt data
```

Рисунок 6 – Расшифруем данные ключом

Заключение

В ходе выполнения лабораторной работы был изучен теоретический материал по шифрованию в PostgreSQL. Приобретенные знания были применены на практике в СУБД PostgreSQL.