Государственное учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Кафедра: Интеллектуальных информационных технологий

Дисциплина: Проектирование защищенных интеллектуальных информационных систем

Отчет по лабораторной работе №3

**Анализ защищенности приложений.**

**Оценка защищенности разработанного приложения**

Выполнил:

студент гр.121702

Витковская С. И.

Проверил:

Захаров В. В.

Минск 2024

**Задание:**

1. Разработать приложение, которое позволяет

- добавить пользователя в систему,

- авторизовать пользователя на основе идентификационных данных,

- создать, редактировать, удалить, осуществить поиск конфиденциальных данных,

- создать, редактировать, удалить, осуществить поиск неконфиденциальных данных,

- деавторизовать авторизованного пользователя.

2. Выполнить анализ разработанного приложения с точки зрения

- политики безопасности операционной системы,

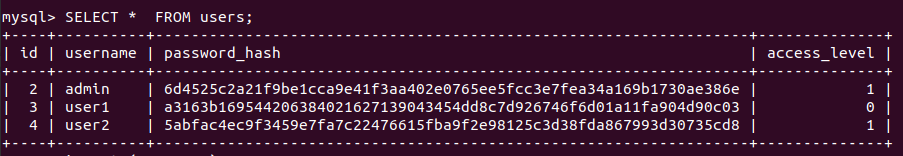
- политики безопасности компонентов приложения и их взаимодействия между собой,

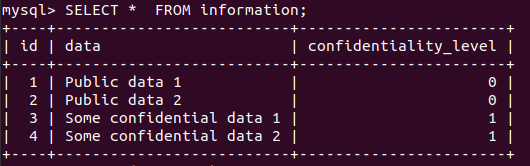
- сетевой политики безопасности,

- безопасности конфиденциальности данных.

**Ход работы:**

1. **Структура БД:**Используемая СУБД – MySQL.

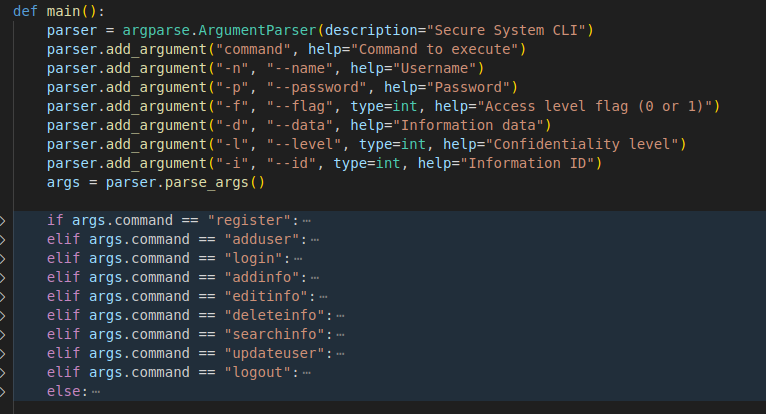
БД содержит таблицы пользователей и таблицу с информацией. Среди пользователей есть администратор, который помимо обладания всеми правами доступа, может добавлять и удалять пользователей, редактировать их права доступа. Взаимодействовать с конфиденциальной информацией могут только пользователи, которых администратор наделил соответствующими привилегиями.  
Тестовая таблица пользователей:  


Тестовая таблица с информацией: 

1. **Описание программы**

Программа представляет собой консольное приложение, взаимодействие с системой производится с помощью команд, с указанием флагов:

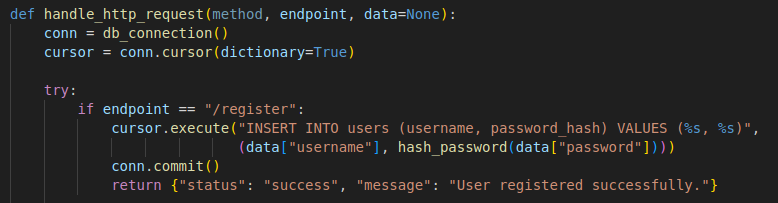
| Регистрация пользователя | register -n "name" -p "password" |
| --- | --- |
| Добавление пользователя (администратор) | аdduser -n "name" -p "password" -f 1 |
| Авторизация | login -n "name" -p "password" |
| Добавление данных | addinfo -d "data" -l 1 |
| Редактирование | editinfo -i 1 -d "new\_data" -l 0 |
| Удаление | deleteinfo -i 1 |
| Поиск | searchinfo -f 1 |
| Обновление прав пользователя | updateuser -n "name" -f 1 |
| Деавторизация | logout |



Программа имитирует веб-приложение, преобразовывая команды в HTTP запросы.



А их, в свою очередь – в запросы к БД.

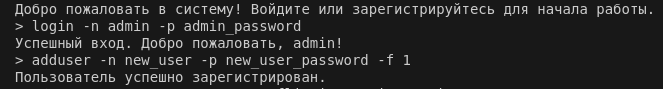


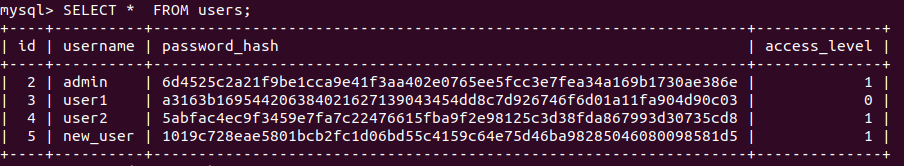
При добавлении пользователя, его пароль сохраняется в БД в хэшированном виде:



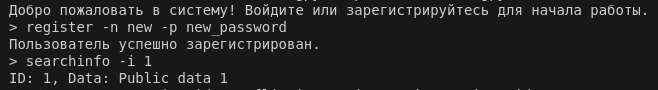
1. **Пример работы программы:**

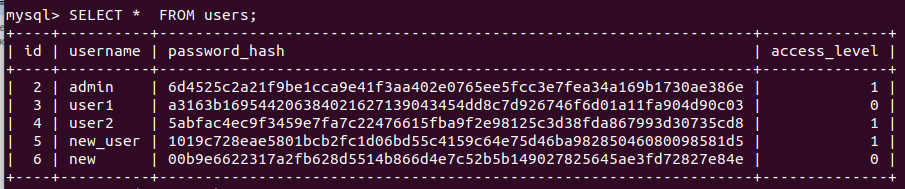
Авторизация администратора и добавление нового пользователя:



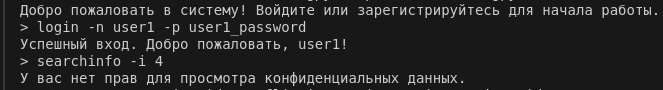


Регистрация пользователя и поиск информации:

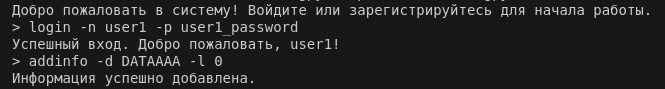


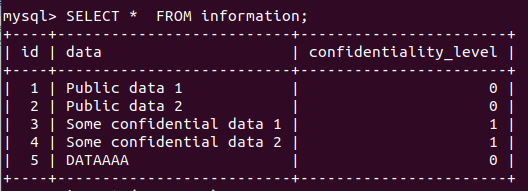


Поиск информации вне прав доступа



Добавление информации





1. **Анализ**
2. Политика безопасности операционной системы

Linux имеет строгую модель управления правами доступа (модель Unix-подобных разрешений), которая ограничивает права пользователей и процессов.

Механизмы вроде SELinux (Security-Enhanced Linux) и AppArmor предоставляют дополнительный уровень контроля, позволяя задавать детализированные политики безопасности для процессов и файлов. Это защищает систему от попыток взлома, даже если злоумышленнику удалось получить доступ. SELinux содержит все необходимое для защиты ОС и приложений от несанкционированного доступа. AppArmor – решение, обладающее половиной функционала SELinux, но гораздо более простое в настройке и эксплуатации. Так же, как и SELinux AppArmor является Обе модели являются реализацией системы Mandatory Access Control (MAC), основанной на архитектуре Linux Security Modules (LSM).

В Linux доступны средства для шифрования данных, такие как LUKS (Linux Unified Key Setup). Пользователь может настроить автоматическое шифрование при установке большинства современных дистрибутивов.

1. Политика безопасности компонентов приложения:

Используются разные уровни доступа. В рамках приложения есть три роли пользователей: admin, user и guest. Каждой роли доступны свои привилегии.

Имитация веб-приложения использует разделение логик HTTP-обработчика и взаимодействия с базой.

1. Сетевая политика безопасности

В рамках приложения используется протокол обмена данными HTTP, по которому можно пересылать любую информацию, но опасность в том, что нет никаких средств для регулирования содержимого HTTP - сообщений.

1. Конфиденциальность данных

Реализован механизм контроля доступа на основе флага доступа.

Пароли всех пользователей хэшируются с помощью SHA-256 и хранятся в зашифрованном виде. А при аутентификации пароль шифруется и проверяется с паролем, сохраненным в памяти приложения.

**Вывод:**

В ходе лабораторной работы была разработана программа для управления доступом к конфиденциальной и неконфиденциальной информации. Используемая СУБД – MySQL. Реализованы механизмы проверки прав доступа пользователей, разделенных на 3 категории: админ, пользователи с доступ к конф. инф-ции, пользователи без данного доступа.