Государственное учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Кафедра: Интеллектуальных информационных технологий

Дисциплина: Проектирование защищенных интеллектуальных информационных систем

Отчет по лабораторной работе №4

**Анализ защищенности приложений.**

**Оценка защищенности сторонних программных систем**

Выполнил:

студент гр.121702

Витковская С. И.

Проверил:

Захаров В. В.

Минск 2024

**Задание:**

1. Разработать методику оценки защищенности систем.

2. Выполнить анализ предоставленного приложения с позиций

- политики безопасности операционной системы,

- политики безопасности компонентов приложения и их взаимодействия между собой,

- сетевой политики безопасности,

- безопасности конфиденциальности данных.

3. Предложить комплекс мер по усовершенствованию приложения.

4. Предложить рекомендации по безопасному использованию приложения.

**Ход работы:**

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.

1. **Комплекс мер по усовершенствованию защиты приложения**
2. Усилить алгоритм шифрования паролей, например:

– bcrypt, добавляющий адаптивный фактор сложности. Хэш включает соль и фактор сложности (`cost`), который определяет количество итераций.

– Argon2 – современный алгоритм с настраиваемыми параметрами: время выполнения (число итераций), объем используемой памяти, количество параллельных потоков.

1. Исключить сохранение паролей или конфиденциальных данных в логах.
2. Не только установить права доступа к данным, но и определить права доступа к самой БД.
3. Добавить другие этапы для аутентификации пользователя, например введение проверки с помощью кода, отправляемого в сообщении или на почту.
4. Валидация ввода:для проверки входных данных на предмет SQL-инъекций.
5. Хранение не только паролей, но и конфиденциальных данных в шифрованном виде.
6. Установить условия для создания пароля, например минимальная длина, наличие специальных символов, чисел и т.д.
7. **Рекомендации по безопасному использованию приложения**

1. Использование сложных паролей: рекомендуется делать длинные пароли с наличием разных типов символов: строчные и заглавные буквы, цифры, специальные символы.

2. Исключить хранение паролей в облаке.

3. Проверка запрашиваемых приложением разрешений при установке.

4. Не вводить конфиденциальные данные в приложении,во время подключения к незащищенным Wi-Fi-сетям (например, в кафе или аэропортах).

**Вывод:**

В ходе лабораторной работы была проанализирована программа из лабораторной работы 2.1. Были выделены преимущества и недостатки, даны рекомендации к усовершенстованию программы и безопасному ее использованию.