**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторному практикуму № 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***на тему*** |  | *Безопасность клиентских операционных систем.* | | |
|  |  | *(наименование темы)* | | |
|  |  |  | | |
| **по дисциплине** | | |  | **Встроенные средства защиты баз данных** |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Пехтерев Алексей Александрович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ВБИо-304рсоб |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Малыш Анатолий Евгеньевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г.**

**Задание 1**

1. Выбрать СУБД, отвечающие интересам компании.
2. Спланировать и описать области проверки безопасности СУБД, используемых в компании.
3. Провести проверку безопасности СУБД, используемых в компании.
4. Сформировать отчет по результатам протоколирования и аудита СУБД.

**Задание 2**

1. Выбрать СУБД, отвечающие интересам компании.
2. Выявить и проанализировать встроенные механизмы защиты, обеспечивающие:

* *конфиденциальность:*обеспечение пользователям доступа только к тем данным, для которых пользователь имеет явное или неявное разрешение на доступ;
* *целостность:*обеспечение защиты от преднамеренного или непреднамеренного изменения информации или процессов ее обработки;
* *доступность:*обеспечение возможности авторизованным в системе пользователям доступа к информации в соответствии с принятой технологией.

1. Оценить эффективность встроенных средств защиты с точки зрения противодействия потенциальным угрозам.

**Задание 3**

Разработайте таблицы определения идентификаторов 3-4 пользователей с заданными вариантами прав. Для практической реализации можно использовать СУБД Access, My SQL или  SQL Server.

**Задание 4**

Для созданного или готового набора таблиц БД (не менее 3-х) Сформулируйте и разработайте не менее 3-х запросов, содержащих выборку из всех таблиц и отражающих только информацию, необходимую пользователю в соответствии с его правами. Для практической реализации можно использовать СУБД Access, My SQL или  SQL Server.

**Задание 1**

1. Выбрать СУБД, отвечающие интересам компании.
2. Спланировать и описать области проверки безопасности СУБД, используемых в компании.
3. Провести проверку безопасности СУБД, используемых в компании.
4. Сформировать отчет по результатам протоколирования и аудита СУБД.

**1. Выбор СУБД, отвечающих интересам компании**

Выбор системы управления базами данных (СУБД) зависит от множества факторов, включая тип данных, объем, требования к производительности, доступности, функциональным возможностям и другим критериям.

Примеры популярных СУБД:

- PostgreSQL — открытая и очень мощная объектно-реляционная СУБД, подходящая для сложных запросов и больших объемов данных.

- MySQL — широко используемая реляционная СУБД, известная своей простотой и хорошей производительностью.

- Microsoft SQL Server — коммерческая СУБД с широкими возможностями для аналитики и интеграции с другими продуктами Microsoft.

- MongoDB — NoSQL СУБД, подходящая для работы с неструктурированными данными и большим объемом информации.

- Oracle DB — мощная коммерческая СУБД, подходящая для крупных корпоративных приложений.

**2. Спланировать и описать области проверки безопасности СУБД**

Проверка безопасности СУБД должна охватывать несколько ключевых областей:

*- Аутентификация и авторизация:*

- Проверка систем аутентификации пользователей и прав доступа

- Настройка многофакторной аутентификации

- Регулярные проверки прав доступа пользователей.

*- Шифрование данных:*

- Проверка наличия шифрования на уровне базы данных и шифрования данных в transit (учетного и другого)

- Использование безопасных протоколов для передачи данных (например, SSL/TLS).

*- Протоколирование и аудит:*

- Ведение журналов активности пользователей (вход, изменения данных, ошибки доступа или попытки несанкционированного доступа).

- Регулярный аудит журналов для обнаружения потенциальных угроз.

*- Защита от SQL-инъекций:*

- Проверка скриптов на наличие уязвимостей

- Настройка средств защиты (например, WAF или использование ORM).

*- Обновления и патчи:*

- Убедиться в регулярном обновлении СУБД и использовании последних версий патчей безопасности.

*- Резервное копирование и восстановление:*

- Проверка наличия актуальных резервных копий базы данных и тестов на восстановление.

**3. Проведение проверки безопасности СУБД**

Для проведения проверки безопасности можно использовать следующие шаги:

1. Аудит политики безопасности: Проверьте существующие политики безопасности и соответствие стандартам (например, GDPR, HIPAA, PCI DSS).

2. Использование инструментов для проверки уязвимостей: Используйте инструменты, такие как:

- Nessus или OpenVAS для проверки уязвимостей

- SQLMap для проверки на SQL-инъекции

- Burp Suite для диагностики веб-приложений.

3. Тестирование на проникновение: Проведите тесты на проникновение, чтобы выявить возможные уязвимости и слабые места в защите.

4. Проверка настройки безопасности: Проанализируйте текущие настройки безопасности СУБД, такие как настройки доступа, шифрование, учетные политики и т. д.

5. Анализ журналов: Просмотрите журналы аутентификации и аудита на предмет подозрительной активности.

**4. Формирование отчета по результатам протоколирования и аудита СУБД**

Отчет должен содержать следующие секции:

*- Введение:*

- Цели и задачи проверки.

- Обозначение СУБД, которую проверяли.

*- Методология:*

- Описание используемых методов и инструментов для проверки безопасности.

*- Обозначение уязвимостей:*

- Перечень выявленных уязвимостей и проблем с безопасностью.

- Описание их серьезности (высокий, средний, низкий риск).

*- Рекомендации:*

- Предложения по решению проблем и повышению уровня безопасности (например, обновление, исправление конфигурации, изменение политик доступа).

*- Заключение:*

- Обобщение результатов проверки.

- Прогноз о состоянии безопасности в будущем и необходимость регулярных проверок.

*- Приложения:*

- Логи и остальные документы, использованные в процессе анализа.

Создание отчетов и документооборота является важной частью обеспечения безопасности данных, поскольку позволяет отслеживать изменения и предпринимать соответствующие меры.

**Задание 2**

Выбрать СУБД, отвечающие интересам компании.

Выявить и проанализировать встроенные механизмы защиты, обеспечивающие:

* *конфиденциальность:*обеспечение пользователям доступа только к тем данным, для которых пользователь имеет явное или неявное разрешение на доступ;
* *целостность:*обеспечение защиты от преднамеренного или непреднамеренного изменения информации или процессов ее обработки;
* *доступность:*обеспечение возможности авторизованным в системе пользователям доступа к информации в соответствии с принятой технологией.

Оценить эффективность встроенных средств защиты с точки зрения противодействия потенциальным угрозам.

При выборе системы управления базами данных (СУБД) для компании важно учесть ее потребности, объемы данных, требования к производительности и интеграции, а также безопасность. Рассмотрим несколько популярных СУБД и их встроенные механизмы защиты.

**Выбор СУБД**

1. PostgreSQL:

- Открытая объектно-реляционная СУБД, поддерживающая расширенные функции.

- Хорошо подходит для обработки больших объемов данных и многопользовательских систем.

2. MySQL:

- Широко используемая реляционная СУБД, поддерживает различные механизмы безопасности и надежная в работе.

- Идеальна для веб-приложений и небольших/средних бизнесов.

3. Microsoft SQL Server:

- Коммерческая СУБД, обладающая мощными функциями безопасности и управления данными.

- Подходит для крупных организаций с большими объемами данных и сложными требованиями к отчетности.

4. Oracle Database:

- Высокопроизводительная коммерческая СУБД с обширным набором функций для крупных организаций.

- Мощные механизмы защиты и управления данными.

**Механизмы защиты**

*1. Конфиденциальность:*

- Аутентификация: Все вышеперечисленные СУБД поддерживают различные формы аутентификации (логин/пароль, Kerberos, OAuth).

- Управление правами доступа: Настройка ролей и привилегий для пользователей с использованием RBAC (role-based access control) позволяет ограничивать доступ к данным.

- Шифрование данных: Многие СУБД поддерживают шифрование данных на уровне столбцов и файловой системы (например, Transparent Data Encryption в SQL Server и Oracle).

*2. Целостность:*

- Контроль целостности данных: СУБД обеспечивают использование ограничений (constraints), триггеров и транзакций, чтобы гарантировать целостность данных.

- Логирование и аудит: Возможности ведения журналов транзакций и аудита позволяют отслеживать изменения и действия пользователей.

*3. Доступность:*

- Резервное копирование и восстановление: Постоянная доступность данных гарантируется механизмами резервного копирования и восстановления до последней подтвержденной транзакции.

- Репликация и масштабирование: Поддержка репликации и кластеризации обеспечивает отказоустойчивость и балансировку нагрузки.

**Оценка эффективности средств защиты**

*- Противодействие угрозам:*

- Аутентификация и управление доступом позволяют минимизировать риск несанкционированного доступа.

- Механизмы шифрования данных обеспечивают защиту путем избежания компрометации данных при утечках.

- Ведение журнала и аудит действий пользователей позволяет быстро реагировать на инциденты и проводить расследования.

- Регулярное резервное копирование и возможность восстановления данных защищают от потерь, данных из-за сбоев или атак.

*- Рекомендации:*

- Регулярно проверять и обновлять настройки безопасности.

- Обучать сотрудников по вопросам безопасности и доступа к данным.

- Проводить регулярные аудиты защищенности базы данных.

Выбор конкретной СУБД и их защита должны основываются на анализе потребностей компании, а также на понимании и учете потенциальных угроз информационной безопасности.

**Задание 3**

Разработайте таблицы определения идентификаторов 3-4 пользователей с заданными вариантами прав. Для практической реализации можно использовать СУБД Access, My SQL или  SQL Server.

Для разработки таблиц определения идентификаторов пользователей с разными уровнями прав доступа, сначала определим, какие права у нас будут, а затем создадим таблицы для пользователей и их прав. Ниже представлена схема таблиц и SQL-запросы для создания этих таблиц в СУБД (например, MySQL).

*Схема таблиц*

*1. Таблица пользователей (users):*

- user\_id (PRIMARY KEY, INT, AUTO\_INCREMENT) — уникальный идентификатор пользователя.

- username (VARCHAR) — имя пользователя.

- password (VARCHAR) — пароль (очень важно хранить в зашифрованном виде на практике).

- role\_id (INT, FOREIGN KEY) — идентификатор роли, указывающий права пользователя.

*2. Таблица ролей (roles):*

- role\_id (PRIMARY KEY, INT, AUTO\_INCREMENT) — уникальный идентификатор роли.

- role\_name (VARCHAR) — название роли.

- permissions (VARCHAR) — описание прав этой роли (например, "READ, WRITE", "READ").

### Пример SQL-запросов

SQL-запросы для создания таблиц и добавления в них пользователей с различными правами доступа:

*--Создание таблицы ролей*

CREATE TABLE roles (

role\_id INT AUTO\_INCREMENT PRIMARY KEY,

role\_name VARCHAR(50) NOT NULL,

permissions VARCHAR(255)

);

*-- Вставка ролей*

INSERT INTO roles (role\_name, permissions) VALUES

('Admin', 'READ, WRITE, DELETE'),

('Editor', 'READ, WRITE'),

('Viewer', 'READ');

*-- Создание таблицы пользователей*

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(255) NOT NULL,

role\_id INT,

FOREIGN KEY (role\_id) REFERENCES roles(role\_id)

);

*-- Вставка пользователей с разными правами*

INSERT INTO users (username, password, role\_id) VALUES

('admin\_user', 'hashed\_password\_1', 1), -- Admin

('editor\_user', 'hashed\_password\_2', 2), -- Editor

('viewer\_user', 'hashed\_password\_3', 3); -- Viewer

**Примечания**

*1. Пароли:* В реальных приложениях пароли необходимо хранить в зашифрованном виде. Используйте такие функции, как bcrypt, чтобы хешировать пароли перед сохранением в базу данных.

*2. СУБД:* Эти SQL-запросы будут работать в MySQL, но могут потребовать некоторых изменений для других СУБД, таких как Microsoft SQL Server или Access (например, синтаксис для работы с автоинкрементами может отличаться).

*3. Подбор прав доступа:* В данном примере показываются основные уровни доступа. Вы можете адаптировать их под конкретные требования вашей системы, расширяя и изменяя роли и их права.

**Задание 4**

Для созданного или готового набора таблиц БД (не менее 3-х) Сформулируйте и разработайте не менее 3-х запросов, содержащих выборку из всех таблиц и отражающих только информацию, необходимую пользователю в соответствии с его правами. Для практической реализации можно использовать СУБД Access, My SQL или  SQL Server.

Для данной задачи мы будем использовать ранее созданные таблицы users и roles. Добавим третью таблицу, например, documents, чтобы работать с запросами. Таблица будет содержать информацию о документах, а также информацию о том, кто их создал и к каким правам у пользователя есть доступ.

Схема таблицы документов

*1. Таблица документов (documents):*

- document\_id (PRIMARY KEY, INT, AUTO\_INCREMENT) — уникальный идентификатор документа.

- document\_name (VARCHAR) — название документа.

- created\_by (INT, FOREIGN KEY) — идентификатор пользователя, создавшего документ.

- content (TEXT) — содержание документа.

- visibility (VARCHAR) — видимость документа (например, "public", "private").

Пример SQL для создания таблицы documents и добавления данных

-- Создание таблицы документов

CREATE TABLE documents (

document\_id INT AUTO\_INCREMENT PRIMARY KEY,

document\_name VARCHAR(100) NOT NULL,

created\_by INT,

content TEXT,

visibility VARCHAR(50),

FOREIGN KEY (created\_by) REFERENCES users(user\_id)

);

*-- Вставка данных в таблицу документов*

INSERT INTO documents (document\_name, created\_by, content, visibility) VALUES

('Document 1', 1, 'Content of Document 1', 'public'), -- Admin created

('Document 2', 2, 'Content of Document 2', 'private'), -- Editor created

('Document 3', 3, 'Content of Document 3', 'public'); -- Viewer created

*Запросы для выборки данных*

Теперь мы сформулируем запросы, которые будут извлекать данные из таблиц в соответствии с правами пользователей.

*Запрос 1*: Получение всех документов для администратора

Этот запрос возвращает все документы, так как у администратора есть полные права.

SELECT

d.document\_id,

d.document\_name,

d.content,

d.visibility

FROM

documents d

JOIN

users u ON d.created\_by = u.user\_id

WHERE

u.username = 'admin\_user';

*Запрос 2:* Получение документов, доступных редактору

Этот запрос возвращает только те документы, которые были созданы редактором или является публичными, то есть с видимостью public.

SELECT

d.document\_id,

d.document\_name,

d.content

FROM

documents d

JOIN

users u ON d.created\_by = u.user\_id

WHERE

u.username = 'editor\_user' OR d.visibility = 'public';

*Запрос 3:* Получение только публичных документов для зрителя

Этот запрос возвращает только те документы, которые имеют публичный доступ.

SELECT

d.document\_id,

d.document\_name,

d.content

FROM

documents d

JOIN

users u ON d.created\_by = u.user\_id

WHERE

d.visibility = 'public' AND u.username = 'viewer\_user';