Лабораторная работа №3
Хеширование паролей

Цель работы: реализовать безопасное хранение паролей с использованием хешей и соли

Ход работы:

Пункт 1. Запуск приложения

```
PS C:\hash_lab\InfoSec-practice-work-03-main> docker compose up -d --build
[+] Building 5.1s (12/12) FINISHED
 => [internal] load local bake definitions                                                                0.1s
 => => reading from stdin 555B                                                                            0.1s
 => [internal] load build definition from Dockerfile                                                      0.0s
 => => transferring dockerfile: 187B                                                                      0.0s
 => [internal] load metadata for docker.io/library/node:24-alpine                                         3.6s
 => [internal] load .dockerignore                                                                         0.0s
 => => transferring context: 2B                                                                           0.0s
 => [1/5] FROM docker.io/library/node:24-alpine@sha256:7e0bd0460b26eb3854ea5b99b887a6a14d665d14cae694b78ae2936d14  0.1s
 => => resolve docker.io/library/node:24-alpine@sha256:7e0bd0460b26eb3854ea5b99b887a6a14d665d14cae694b78ae2936d14  0.1s
 => [internal] load build context                                                                         0.0s
 => => transferring context: 402B                                                                         0.0s
 => CACHED [2/5] WORKDIR /app                                                                             0.0s
 => CACHED [3/5] COPY package.json yarn.lock ./                                                           0.0s
 => CACHED [4/5] RUN yarn install --frozen-lockfile                                                       0.0s
 => [5/5] COPY . .                                                                                        0.1s
 => exporting to image                                                                                    0.6s
 => => exporting layers                                                                                   0.3s
 => => exporting manifest sha256:5653672252427261c15c21cb95858aae4990d15ee51566475d087c026ffbf4bc         0.0s
 => => exporting config sha256:3adecd3c4252c78d672457975d5226537e6fe54691a2ff42521bce2737948e09           0.0s
 => => exporting attestation manifest sha256:82e7e341a83f01a4f5c66c90b4284b87b5a41c9dfa4c06b60ff1030a2bf8517b  0.0s
 => => exporting manifest list sha256:791267c03b729cbaac0c9ec2312257e50c86e12b83716332991dbc645a3636b5    0.0s
 => => naming to docker.io/library/api_image:latest                                                       0.0s
 => => unpacking to docker.io/library/api_image:latest                                                    0.1s
 => resolving provenance for metadata file                                                                0.1s
[+] Running 5/5
 ✔ api_image:latest                                Built                                                  0.0s
 ✔ Network infosec-practice-work-03-main_default   Created                                                0.1s
 ✔ Container db                                    Started                                                3.0s
 ✔ Container pg                                    Started                                                3.3s
 ✔ Container api                                   Started                                                3.4s
```

Результат: Контейнеры успешно собраны и запущены.

Пункт 2. Проверка работы приложения

```
PS C:\hash_lab\InfoSec-practice-work-03-main> curl.exe --location 'http://localhost:3000/health'
{"status":"ok"}
```

Результат: `{"status":"ok"}` — сервер работает.

Пункт 3. Создание пользователя (до модификации)

```
PS C:\hash_lab\InfoSec-practice-work-03-main> curl.exe --location 'http://localhost:3000/register' --header 'Content-Type: application/json' --data-raw "{\"username\": \"test@example.com\", \"password\": \"super_secret_passw
ord\"}"
{"id":1,"username":"test@example.com","password":"super_secret_password","updatedAt":"2025-12-11T23:30:11.410Z","createdAt":"2025-12-11T23:30:11.410Z","salt":null}
                oSec-practice-work-03-main> docker compose restart
 ✔ Container db   Started                                                                                 1.1s
 ✔ Container api  Started                                                                                 0.0s
 ✔ Container pg   Started                                                                                 1.0s
```

Вывод: Пароль сохраняется в открытом виде.

4 и 8 отправим запрос на проверку работы программы:

Пункт 4. Проверка данных в БД (до модификации)

```
PS C:\hash_lab\InfoSec-practice-work-03-main> docker exec -it db psql -U student -d db -c "SELECT * FROM public.users;"
 id |      username      |                           password                           | salt |       createdAt        |       updatedAt
----+--------------------+--------------------------------------------------------------+------+------------------------+------------------------
  1 | test@example.com   | super_secret_password                                        |      | 2025-12-11 23:30:11.41+00 | 2025-12-11 23:30:11.41+00
```

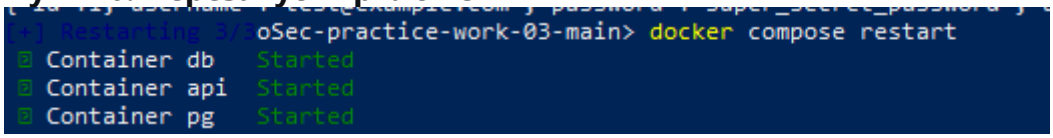
Пункт 5. Модификация кода `server.js`

Добавлено:

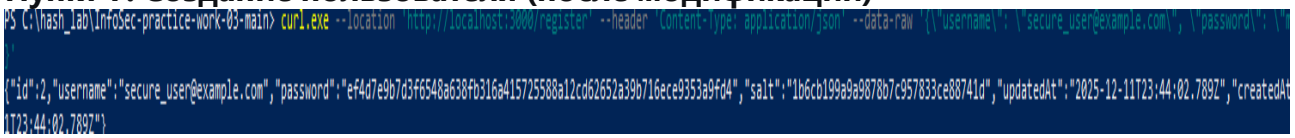
```
import crypto from 'crypto';

function generateSalt() { return crypto.randomBytes(16).toString('hex'); }
function hash(password, salt) { return crypto.createHash('sha256').update(password + salt).digest('hex'); }
```

Изменён маршрут `/register` — теперь пароль хешируется с солью.

Пункт 6. Перезапуск приложения

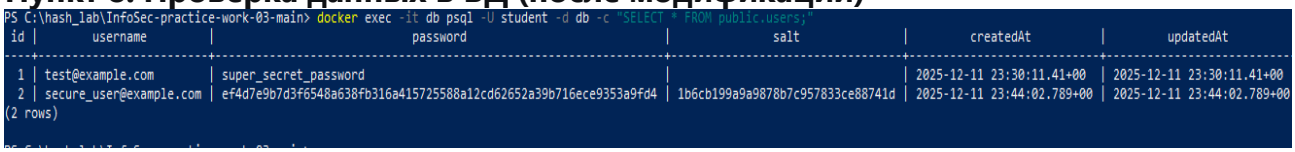


Пункт 7. Создание пользователя (после модификации)



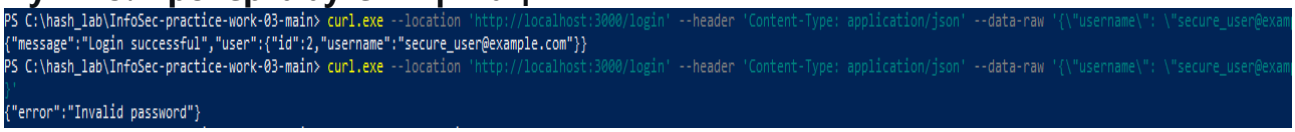
Вывод: Пароль сохранён как хеш SHA-256, соль сгенерирована.

Пункт 8. Проверка данных в БД (после модификации)



Вывод: Второй пользователь имеет захешированный пароль и уникальную соль.

Пункт 9. Проверка аутентификации



Пункт 10. Выводы

1. Реализовано безопасное хеширование паролей с использованием SHA-256 и соли.
2. До модификации пароли хранились открыто, что небезопасно.
3. После модификации каждый пароль хешируется с уникальной солью.
4. Система аутентификации корректно проверяет хеши паролей.
5. Хеширование защищает данные даже при утечке базы данных.