При выполнении лабораторной работы руководствовался статьей Password Storage Cheat Sheet для выбора алгоритмов и способов хранения пароля.

1. Password hashing pipeline

Процес хешериования пароля состоит из 3 частей: pre-hashing password, password hashing, post-hashing password encryption.

Pre-hashing password необходим для того чтобы избавиться от проблем связанных с длинами и с кодировкой паролей. Так как часть хеш алгоритмов воспримчивы к размеру входа (ограничения сверху по размеру или долгое время вычесений для длинных сообщений), а так же к входящим символам (Null bytes bcrypt) необходимо сначала захешировать с помощью достаточно быстрого алгоритма и отобразить в безпасной кодировке base64 или hexadecimal. Для таких целей выбрал SHA-512, так как это быстрый и достаточно безопасный алгоритм SHA-2 с большим размером message digests.

Для хешерования необходимо выбрать криптографически стойкий алгоритм с длительным временем вычеслений и солью, чтобы лишить злоумышлеников возможности быстрого параллельного вычисления хешей. Для такого хорошо подходит алгоритм argon2i, который позоволяет задать количество потраченной памяти, количетво потоков и количество итераций.

Шифрование паролей неоходимо чтобы при утечке базы лишить возможности злоумышлеников перебирать пароли до расшифровки. Для этого хорошо подойдут алгоримы AEAD с возможностью верификации данных на их аутентичность для избежания подмены. Для это выбрал алгоритм рекомендованый библиотекой libsoduim XChaCha20-Poly1305, поволяющий шифровать сообщения любой длины используя один ключ длительное время.

2. Security tools

Для генерации случайной соли, а так же для хеширования с помощью SHA-512 использовал модуль NodeJS crypto. Модуль является оберткой поверх OpenSSL. Для хеширования argon2i использовалась библиотека argon2-ffi, которая пробрасывает оргинальную библиотеку argon2 (победитель Password Hashing Competition и рекомендуется OWASP) реализовывающую алгоритм. Для шифрования XChaCha20-Poly1305 использовалась библиотека sodium-native, которая пробрасывает оригинальную реализацию библиотеки libsodium.

3. Additional steps

При создании аккаунта пароли проверяются на:

- минимальная длина 10 символов;
- наличие одной большо буквы, одной маленькой буквы, одной цифры и одного спец символа;
- отсутвие пароля в списке топ миллион паролей.

Для подбора параметров сложности argon2i использовались возможности продукта ORY Kratos, так чтобы среднее время авторизации пользователя длилось 0.5 секунд. А для сравнения паролей использовались встроенные функции, чтобы избежать time attacks.