

## ТЕХНІЧНЕ ЗАВДАННЯ

**Тема:** Розробка програмного забезпечення для розгортки та встановлення безпечного vpn-з'єднання.

**Мета проєкту:** Розробити комплексне програмне рішення, що дозволяє користувачам легко створювати приватні VPN-мережі, керувати доступом користувачів та груп, і автоматично отримувати необхідні конфігураційні файли для підключення. Мета – абстрагуватися від складності ручного налаштування OpenVPN та PKI (Public Key Infrastructure).

### Вимоги:

1. Платформа та сумісність:
  - a. Серверна частина програми повинна бути розроблена для роботи на 64-бітних версіях ОС Linux (розробка ведеться на Arch). У подальшому можна перенести все на Docker.
  - b. Клієнтська частина повинна бути кросплатформенною, з пріоритетною підтримкою 64-бітних ОС Linux та можливістю компіляції та запуску під ОС Windows.
2. Структура:
  - a. “Сервер” реально являє собою декілька серверів та клієнтів:
    1. VPN-сервер
    2. SSL/STL-сервер
    3. SMTP-клієнт
    4. HTTP/S-сервер
    5. Database-клієнт
  - b. Клієнт реально являє собою:
    1. VPN-клієнт
    2. SSL/STL-клієнт
  - c. “Common libs”: ”частини коду”, що використовуватимуться, як на серверній так і клієнтській частині.
3. Технологічний стек:
  - a. Програма повинна бути написана переважно на мові програмування C++ (стандарт 17)
  - b. Серверна частина повинна використовувати бібліотеку Boost.Asio та Boost.Beast для асинхронної реалізації мережевих протоколів (TCP/TLS, HTTP/S), libcurl для SMTP – клієнта.
  - c. Клієнтська частина повинна використовувати Qt6 для написання графічної частини та, можливо, збереження токенів (небезпечно, але підходить для старту).
  - d. Для криптографічних операцій (хешування, генерація токенів) повинна використовуватись бібліотека libsodium.
  - e. Для зберігання даних – Sqlite3.
  - f. Збірка залежностей – Cmake, білд – Ninja.
4. Безпека та мережа:
  - a. Програма повинна забезпечувати створення безпечного VPN-з'єднання на базі OpenVPN

- b. Увесь управляючий трафік між клієнтом та сервером (реєстрація, авторизація, передача конфігурацій) має бути захищений за допомогою шифрування TLS.
  - c. Програма повинна реалізувати систему автентифікації користувачів за логіном (email) та паролем, а також систему сесій на основі Access та Refresh токенів
  - d. Паролі користувачів повинні зберігатися у вигляді захищених хешів (з використанням солі та алгоритму Argon2id13).
5. Функціональність сервера:
- a. Сервер повинен автоматично генерувати унікальні конфігураційні файли (.ovpn) та сертифікати для кожного зареєстрованого та верифікованого користувача.
  - b. Сервер повинен мати можливість створювати ізольовані групи користувачів та динамічно керувати мережевими правилами доступу між ними за допомогою iptables та ipset.
  - c. Сервер повинен підтримувати механізм верифікації користувачів через підтвердження електронної пошти.
6. Функціональність клієнта:
- a. Програма повинна мати графічний інтерфейс користувача (GUI), реалізований на Qt.
  - b. Клієнт повинен надавати інтерфейс для реєстрації, входу та (опціонально) відновлення пароля.
  - c. Клієнт повинен автоматично отримувати та зберігати конфігураційний файл VPN після успішної авторизації.
  - d. Клієнт повинен мати можливість ініціювати та розривати VPNз'єднання, взаємодіючи з локально встановленим процесом OpenVPN.
  - e. Клієнт повинен відображати список груп, до яких належить користувач, та надавати інтерфейс для управління групами (якщо користувач є їх власником).