**PSI Projekt**

**Treść:** Celem projektu jest zaprojektowanie oraz implementacja szyfrowanego protokołu opartego na protokole TCP, tzw. mini TLS.

**Założenia:**
• Architektura klient serwer.
• Serwer jest w stanie obsłużyć kilku klientów jednocześnie (proszę nie hardcodować liczby klientów – oczekuję parametru uruchomienia).
• Klient inicjuje połączenie z serwerem poprzez wysłanie wiadomości ClientHello (nieszyfrowana), na którą serwer odpowiada wiadomością ServerHello (nieszyfrowana).
• Sesja może zostać zakończona zarówno przez klienta jak i przez serwer poprzez wysłanie wiadomości EndSession. Po odebraniu EndSession należy od nowa wysłać ClientHello.
• Wszystko poza ClientHello i ServerHello jest szyfrowane.
• Potencjalny napastnik po przechwyceniu wiadomości nie jest w stanie z nich nic odczytać.

**Wymagania implementacyjne:**
• Wiadomości ClientHello i ServerHello służą do wymiany kluczy szyfrujących. Po wymianie tych wiadomości klient oraz serwer muszą być w posiadaniu tego samego klucza szyfrującego, którym będą szyfrować kolejne wiadomości.
• Użycie algorytmu wymiany kluczy (np. Diffie-Hellman key exchange). Samo szyfrowanie wiadomości może (ale nie musi) być zrealizowane prostym OTP (ang. one time pad).
• Komunikacja ma być sterowana z wiersza poleceń, tj. uruchamiamy serwer, uruchamiamy klienta (bądź kilku) i mamy do wyboru:
    - Serwer:
        - Zakończ połączenie dla wybranego klienta.
    - Klient:
        - Zainicjuj połączenie z serwerem.
        - Wyślij wiadomość (treść jest nieistotna).
        - Zakończ połączenie z serwerem.

Ponadto serwer wyświetla wszystkich obecnie połączonych klientów oraz odbierane wiadomości.
• Komunikacja ma się odbywać w sieci dockerowej.
• Całość powinna dać się uruchomić minimalną liczbą komend.

**Wymagania sprawozdania:**
• Sprawozdanie wstępne:
    - Struktura wiadomości ClientHello, ServerHello, EndSession oraz szyfrowanych wiadomości.
    - Wykorzystane algorytmy, tj. wymiana kluczy, szyfrowanie wraz z przykładowym scenariuszem użycia wykorzystującym strukturę wiadomości.
    - Sposób realizacji mechanizmu integralności i autentyczności dla szyfrowanych wiadomości (jeśli występuje).
• Sprawozdanie końcowe:
    - Dowód, że protokół działa według założeń, tj. wiadomości ClientHello, ServerHello itd. są wysyłane w odpowiedniej kolejności(np. za pomocą wireshark). Proszę także pokazać, że szyfrowane wiadomości faktycznie po odszyfrowaniu są tym czym być powinny – do tego niezbędne będzie zapisanie gdzieś (np. do pliku) użytego klucza szyfrującego i „ręczne” odszyfrowanie tego co pokazał wireshark.
    - Opis użytych algorytmów.
    - Napotkane problemy.
    - Wnioski.

**Warianty funkcjonalne:**
• W1 – wykorzystanie mechanizmu encrypt-then-mac dla wysyłanych szyfrowanych wiadomości jako mechanizm integralności i autentyczności, implementacja w Pythonie.
• W2 – wykorzystanie mechanizmu mac-then-encrypt dla wysyłanych szyfrowanych wiadomości jako mechanizm integralności i autentyczności, implementacja w Pythonie.
• W3 – bez mechanizmu integralności i autentyczności, implementacja w C (tutaj tylko oczekuję opisać w sprawozdaniu co to jest i czym się charakteryzuje encrypt-then-mac i mac-then-encrypt).

**Terminy:**
- 31.12.2025 – oddanie sprawozdania wstępnego.
- 18.01.2026 – oddanie projektu.
Proszę pilnować wyznaczonych terminów. Przekroczenie terminu będzie skutkować obniżeniem ostatecznego wyniku o 5 punktów za każdy dzień. Projekt jest oceniany na maksymalnie 40 punktów. Minimalna liczba punktów do zaliczenia to 20. Niezalicznie projektu (< 20 p.) powoduje niezaliczenie przedmiotu niezależnie od reszty punktacji.

**Uwagi:**
• Wszystko może być przechowywane w pamięci, nie ma potrzeby tracić czasu np. na konfigurację bazy danych.
• W przypadku mechanizmu wymiany kluczy opartego o liczby pierwsze nie muszą to być duże i „bezpieczne” liczby – spokojnie mogą to być małe liczby typu int.
• Proszę postawić na prostotę rozwiązania – zaproponowana struktura wymienianych wiadomości powinna być jak najprostsza i nie powinna posiadać żadnych nadmiarowych pól.
• Zaimplementowany protokół będzie miał mnóstwo luk bezpieczeństwa - proszę nie implementować czegokolwiek (dodatkowych „zabezpieczeń”) co nie jest wymienione w instrukcji.
• Forma oddania projektu i sprawozdania ta sama co w przypadku laboratoriów, tj. GitHub + sprawozdanie pdf + README.

Powodzenia!