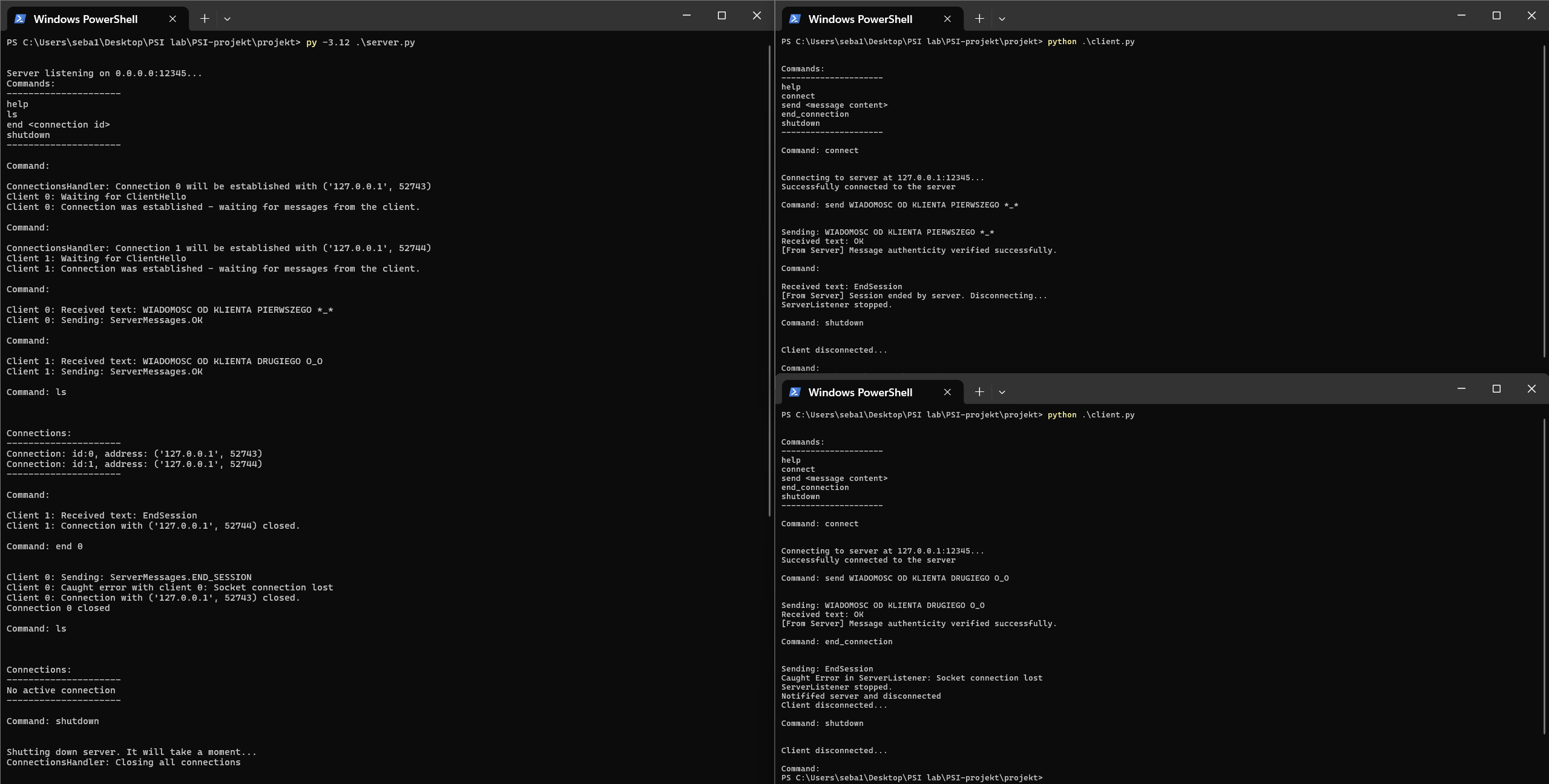
**Sprawozdanie**

**PSI 24Z - Mini TLS**

Wykonali: Sebastian Abramowski, Aleksander Stanoch, Bogumił Stoma

Dnia: 4.01.2025

**Kompleksowe przedstawienie funkcjonalności klienta i serwera**



Serwer oferuje możliwość:

* wyświetlenia pomocy
* wyświetlenia listy aktualnie połączonych klientów
* zakończenia połączenia z danym klientem
* bezpiecznego wyłączenia

Klient pozwala na:

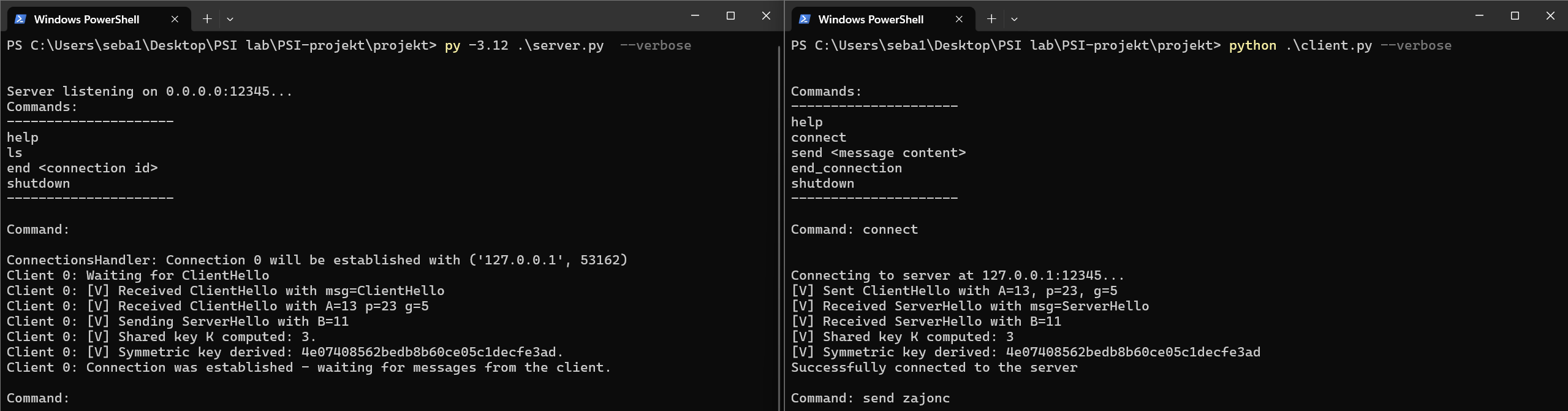
* wyświetlenie pomocy
* połączenie się z serwerem
* wysłanie wiadomości
* zakończenie połączenia
* bezpieczne wyłączenie

Na powyższym zrzucie ekranu widać konsolę serwer (po lewej) oraz 2 konsole klientów (po prawej),

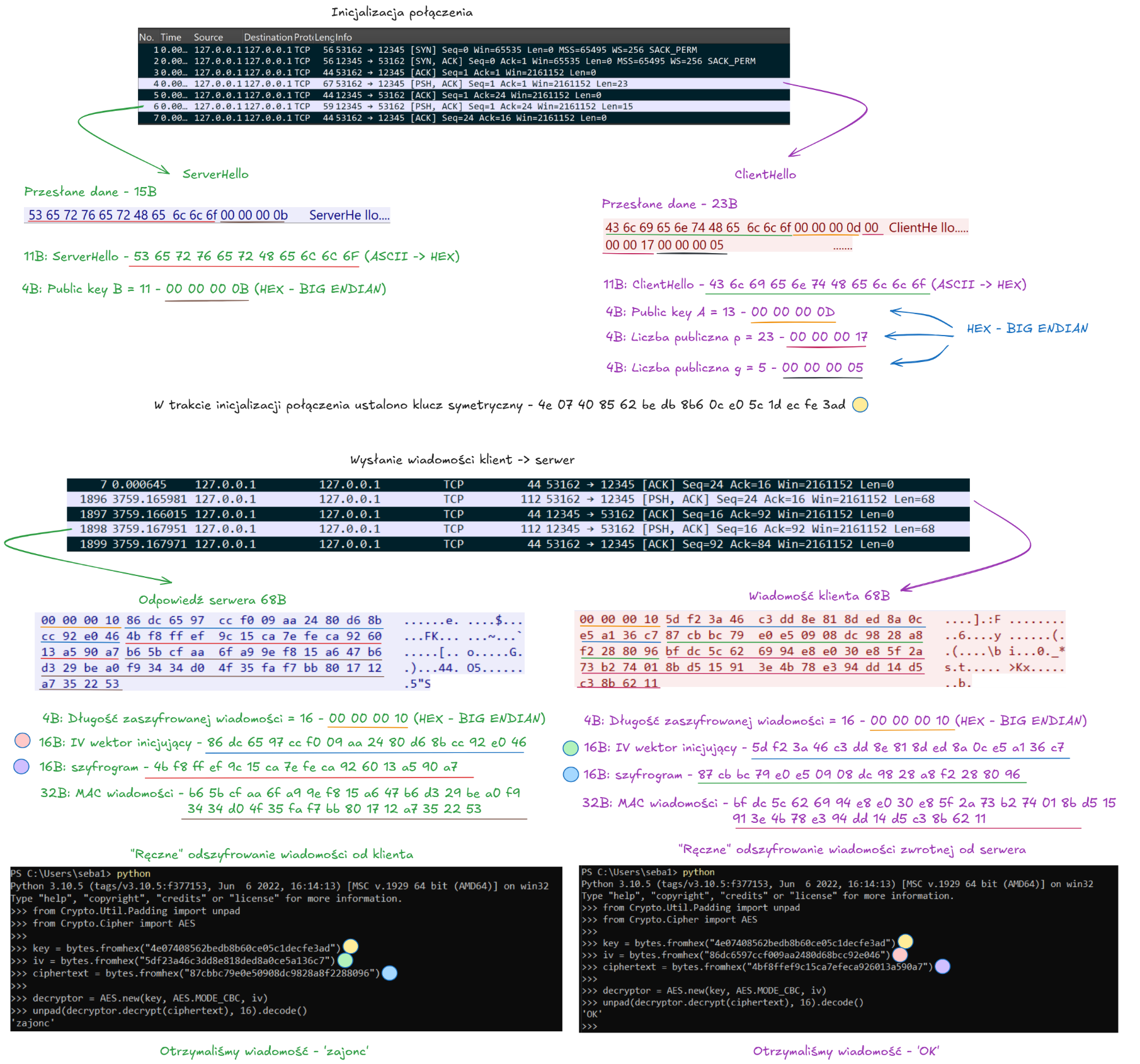
przedstawiają one przykładowy proces wymiany informacji, który przebiegł następująco:

1. Klient 0 (u góry) połączył się z serwerem otrzymując id 0
2. Klient 1 (na dole) połączył się z serwerem otrzymując id 1
3. Klient 0 wysyła widmowość „WIADOMOŚĆ OD KLIENTA PIERWSZEGO”
4. Klient 1 wysyła widmowość „WIADOMOŚĆ OD KLIENTA DRUGIEGO”
5. Serwer wyświetla dostępne połączenia
6. Klient 1 rozłącza się z serwerem
7. Serwer rozłącza klienta 0
8. Serwer wyświetla listę aktywnych połączeń

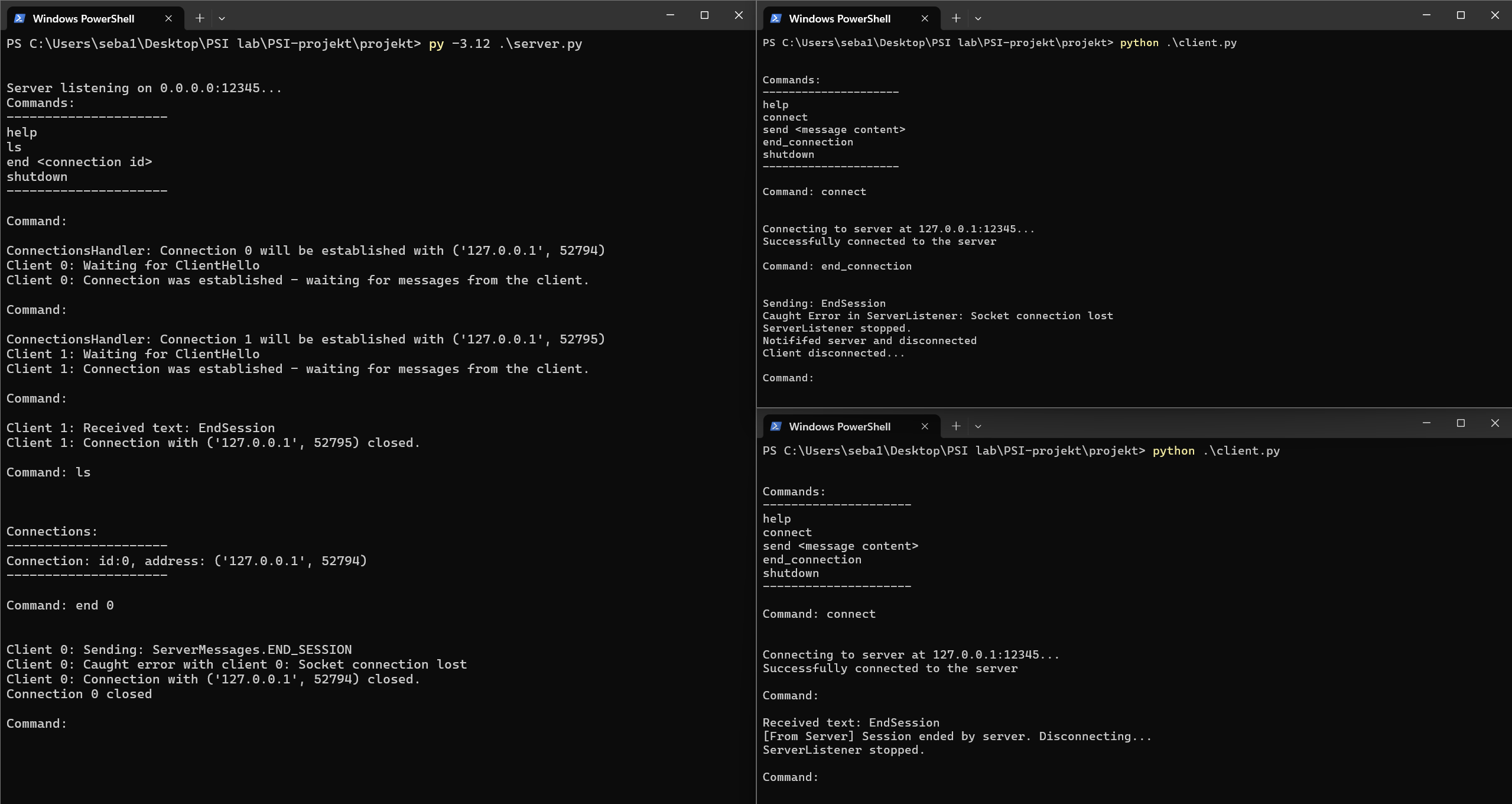
**Zainicjowanie połączenia klienta z serwerem (wiadomości ClientHello i ServerHello)**



Na załączonych zrzutach ekranu uruchomiono programy z funkcją verbose, żeby pokazać jakie wartości zostały wykorzystane do utworzenia bezpiecznego klucza

**Zaobserwowanie wiadomości ClientHello i ServerHello w Wiresharku oraz ‘’ręczne” odczytanie przechwyconych wiadomości**

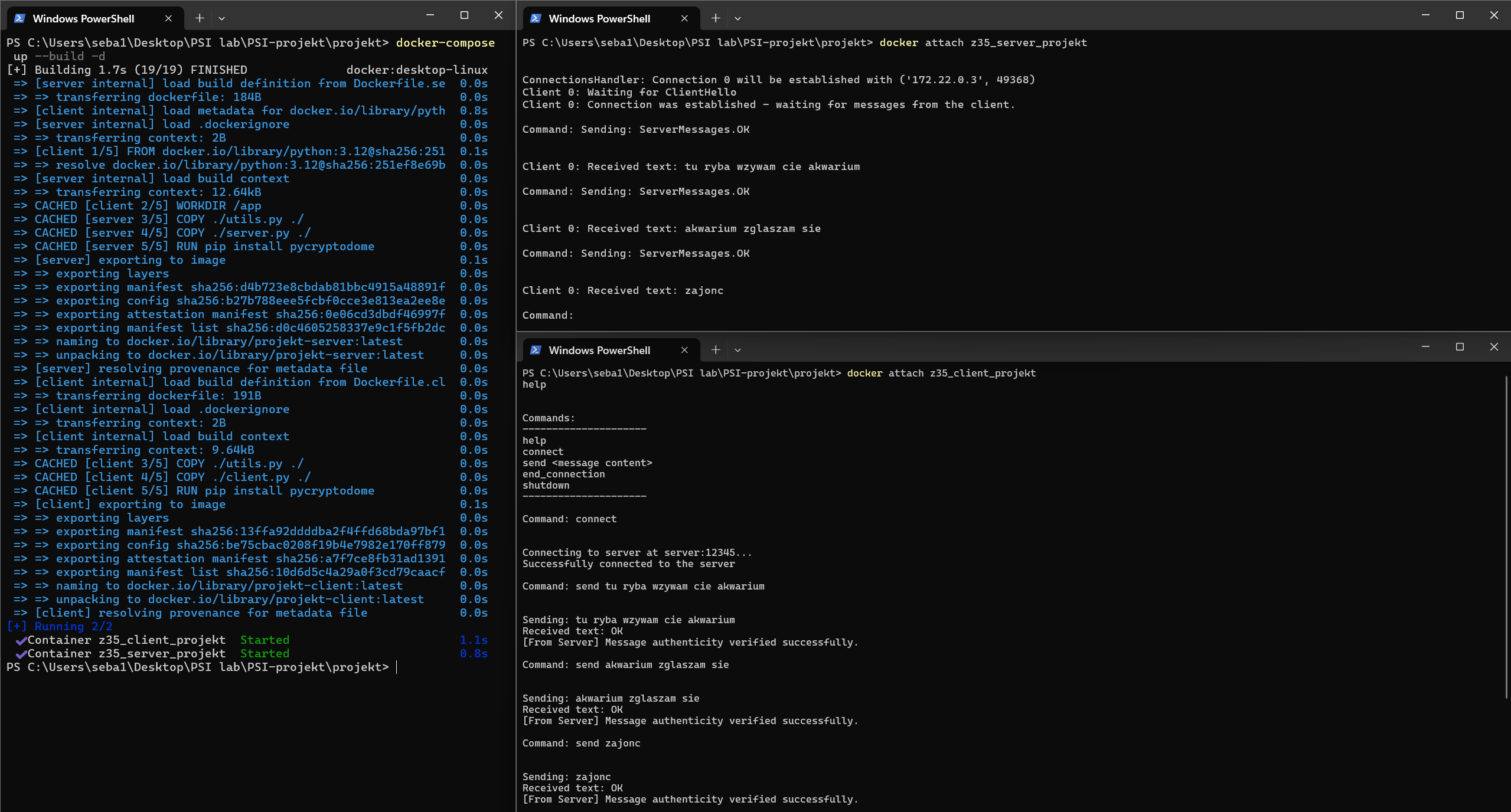
**Zamykanie połączenia przez klienta i serwer**



W konsoli serwera widać że klient 1 (u góry) zażądał zakończenia połączenia wysyłając w zaszyfrowanej wiadomości treść „EndSession” dlatego serwer zakończył połączenia, po czym klient również zakończył połączenie.

Poniżej widać iż serwer zażądał zakończenia połączenia z klientem 0 poprzez wysłanie wiadomości w identyczny sposób, co zarejestrował klient 0 (na dole), kończąc połączenie, na co zareagował serwer rejestrując zerwanie połączenia.

**Działanie w kontenerze**



**Opis użytych algorytmów**

**Wymiana kluczy – Algorytm Diffiego-Hellmana**

* **p** jest dużą liczbą pierwszą,
* **g** jest dowolną liczbę nazywaną generator.
* **a** to wygenerowana prywatna liczba klienta trzymana w tajemnicy.
* **b** to wygenerowana prywatna liczba serwera trzymana w tajemnicy.

1. Klient wysyła swój klucz publiczny **A** (**A** = **g**^**a** mod **p**) wraz z wartościami **p** i **g** za pomocą nieszyfrowanej wiadomości ClientHello.
2. Serwer odpowiada swoim kluczem publicznym **B** (**B** = **g**^**b** mod **p**) za pomocą nieszyfrowanej wiadomości ServerHello.
3. Obie strony obliczają wspólny klucz **K**:
4. Klient: **K** = **B**^**a** mod **p**.
5. Serwer: **K** = **A**^**b** mod **p**.
6. Klucz K będzie używany do symetrycznego szyfrowania komunikacji.

**Szyfrowanie wiadomości – AES w trybie CBC**

* Szyfrowanie:
  + Wejście: wiadomość plaintext, klucz K, wektor inicjalizacyjny (IV).
  + Wyjście: ciphertext (zaszyfrowana treść).
* Odszyfrowanie:
  + Wejście: ciphertext, K, IV.
  + Wyjście: plaintext.

Wektor inicjacyjny (IV) potrzeby jest w algorytmie CBC, jako startowa wartość. Algorytm CBC dodatkowo szyfruje kolejne bloki na które został podzielony tekst operacją XOR z wartością poprzedniego bloku. W ten sposób nawet identyczne znaki zostaną zakodowane w inny sposób, ma to na celu uniemożliwienie złamania kodu bez znajomości całości wiadomości.

**Mechanizm Encrypt-then-MAC**

* Po zaszyfrowaniu wiadomości z ciphertextu oraz klucza symetrycznego K generowany jest kod HMAC przy użyciu SHA-256.
* Wiadomość przesyłana do odbiorcy zawiera:
  + Długość Ciphertext
  + IV
  + Ciphertext.
  + MAC.
* Odbiorca w analogiczny do nadawcy sposób wylicza kod HMAC używając swojego klucza K i otrzymanego Ciphertext, jeśli wygenerowany kod jest zgodny z otrzymanym wiadomość jest poprawna.

**Napotkane problemy**

Napotkaliśmy drobne trudności z synchronizacją wątków, szczególnie w kontekście poprawnego wypisywania komunikatów na konsolę w środowisku wielowątkowym.

**Wnioski**

1. **Poprawność działania**: Protokół Mini TLS działa zgodnie z założeniami. Każdy etap komunikacji został potwierdzony za pomocą Wireshark i manualnego odszyfrowania wiadomości.
2. **Bezpieczeństwo**: Zastosowanie mechanizmu Encrypt-then-MAC zapewnia integralność i autentyczność wiadomości. Algorytmy Diffiego-Hellmana i AES gwarantują poufność transmisji.