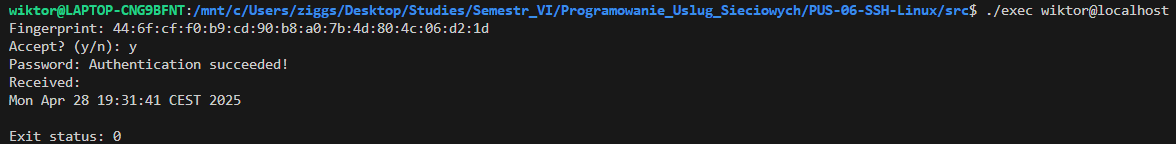
**SSH od strony programistycznej  
Laboratorium 6  
Wiktor Zmiendak**

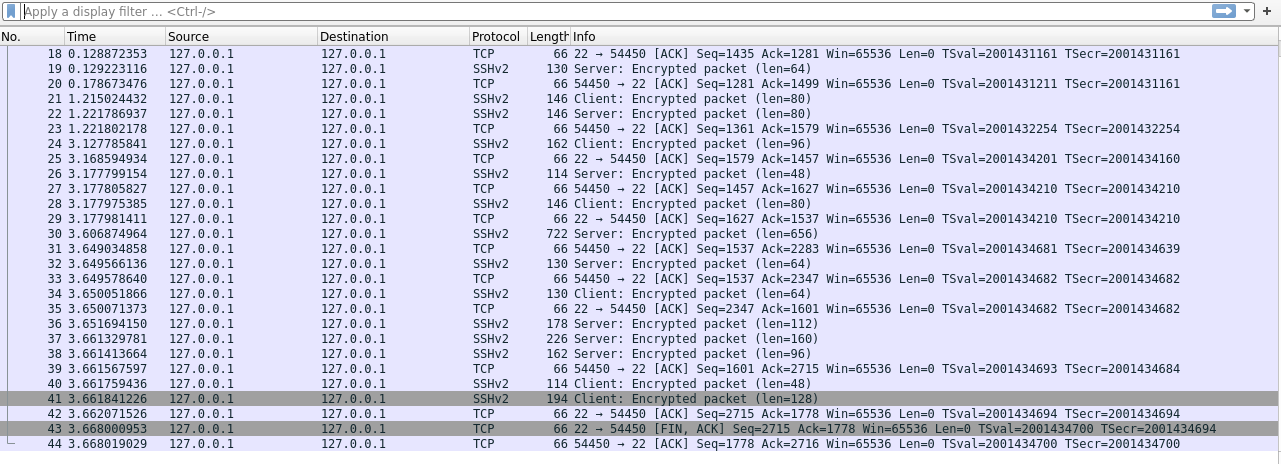
**Obraz zawierający tekst, numer, Czcionka, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.ZADANIE 1**

Program authlist używa biblioteki libssh2 do połączenia z serwerem SSH i pobrania listy dostępnych metod uwierzytelniania dla podanego użytkownika za pomocą funkcji libssh2\_userauth\_list. Analiza kodu i protokołu SSH (RFC 4252) pokazuje, że serwer odpowie listą metod nawet dla fikcyjnej nazwy użytkownika. Dzieje się tak, ponieważ serwer zwraca listę dostępnych opcji w odpowiedzi na pierwszą, zazwyczaj nieudaną próbę uwierzytelnienia (np. metodą "none"), zanim szczegółowo zweryfikuje samo istnienie konta. Przechwycona komunikacja sieciowa potwierdza prawidłowe nawiązanie połączenia i rozpoczęcie fazy uwierzytelniania, w której ta wymiana ma miejsce.

**ZADANIE 2**

****

****

Wywołanie polecenia zdefiniowanego w kodzie źródłowym programu daje jako rezultat wyświetlenie na lokalnym terminalu aktualnej daty i czasu systemowego zdalnego serwera SSH, z którym nawiązano połączenie. Program drukuje ten wynik poprzedzony napisem "Received:"

**ZADANIE 3**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Celem zadania było zmodyfikowanie istniejącego programu C tak, aby realizował uwierzytelnianie użytkownika na serwerze SSH za pomocą metody klucza publicznego, wykorzystując bibliotekę libssh2. Zgodnie z poleceniem, podjęto próbę użycia pary kluczy RSA zabezpieczonej hasłem. W trakcie implementacji napotkano jednak trudności z poprawnym uwierzytelnieniem przy użyciu tej pary kluczy w programie mimo iż standardowy klient OpenSSH działał poprawnie. Dodatkowo, ograniczenia systemu plików NTFS w WSL uniemożliwiły ustawienie wymaganych, restrykcyjnych uprawnień dla klucza prywatnego przechowywanego w katalogu projektu (/mnt/c/...). W celu obejścia napotkanych problemów, wygenerowano alternatywną parę kluczy typu Ed25519 bez hasła za pomocą ssh-keygen i umieszczono ją w standardowej lokalizacji ~/.ssh/ w systemie plików WSL.

**Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

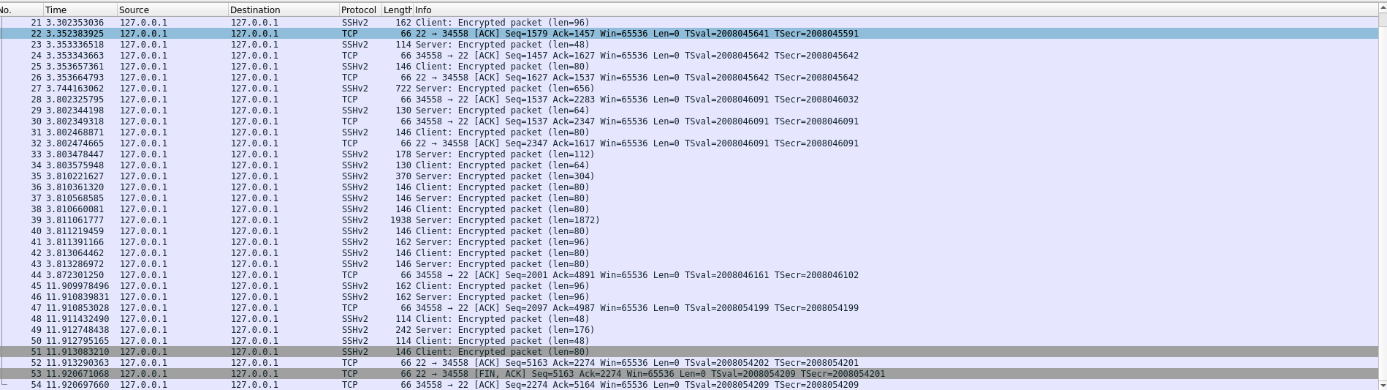
Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.ZADANIE 4**

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

Program poprawnie inicjuje połączenie TCP, sesję SSH oraz podsystem SFTP. Uwierzytelnianie użytkownika realizowane jest dwuetapowo: najpierw podejmowana jest próba autoryzacji za pomocą klucza publicznego, przy czym ścieżki do plików kluczy oraz hasło do klucza prywatnego są na stałe wpisane w kodzie wywołującym funkcję uwierzytelniającą. W przypadku niepowodzenia tej metody, program przechodzi do interaktywnego uwierzytelniania hasłem, pytając użytkownika o nie w pętli aż do uzyskania poprawnej autoryzacji. Po pomyślnym uwierzytelnieniu, program listuje zawartość katalogu domowego i prawidłowo zamyka połączenie oraz zwalnia zasoby.

**ZADANIE 5**

****

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

Zmodyfikowany program, po pomyślnym połączeniu, uwierzytelnieniu i wyświetleniu listy plików w katalogu /home/username, pyta użytkownika o nazwę pliku do pobrania. Odczytana nazwa jest używana do skonstruowania pełnej ścieżki zdalnej i otwarcia zdalnego pliku za pomocą libssh2\_sftp\_open\_ex. Następnie program otwiera plik lokalny o tej samej nazwie w bieżącym katalogu roboczym i w pętli kopiuje zawartość pliku zdalnego do lokalnego, używając funkcji libssh2\_sftp\_read oraz fwrite. Po zakończeniu transferu pliki są zamykane, a użytkownik informowany o statusie pobierania i liczbie pobranych bajtów.