

1. WSTĘP

Projekt realizuje autoryzację metodą „UDP port knocking”, po czym nawiązuje połączenie TCP z klientem.

2. ARCHITEKTURA

Serwer:

1) Start

Serwer uruchamiany jest poprzez wywołanie metody `main()`, która jako argument przyjmuje listę portów, na których serwer ma nasłuchiwać. Dla każdego z podanych portów zostanie utworzony obiekt klasy `ServerPort`, który będzie odpowiedzialny za obsługę tego portu.

2) Autoryzacja

Logika serwera jest odpowiedzialna za przekazywanie pakietów z `ServerPort` bezpośrednio do klasy serwera w celu sprawdzenia ich poprawności. Obiekty `ServerPort` będą przekazywać odebrane pakiety UDP do odpowiednich kolekcji, wątek serwera w tym czasie będzie pobierał te pakiety od obiektów `ServerPort` i kiedy liczba otrzymanych pakietów będzie się zgadzała z liczbą otwartych portów dla pakietów UDP to uruchomi wątek sprawdzający ich poprawność, jeżeli pakiety utworzyły listę kolejnych liter alfabetu (początkowa litera zawsze jest losowa), proces ten wyśle losowo wygenerowany port na którym serwer będzie oczekiwał połączenia TCP na adres IP z którego przychodziły pakiety i na port ustalony z góry przez serwer. Jeżeli sekwencja się nie zgadza wątek sprawdzający poprawność sekwencji skończy pracę.

3) Otworzenie połączenia TCP

Po udanej autoryzacji proces sprawdzania poprawności pakietów otworzy port wysłany pakietem do klienta, który będzie czekał na połączenie TCP, kiedy nastąpi połączenie wątek połączenia TCP wyśle pakiet w którym będzie zawarte słowo „REQUEST” do klienta i po otrzymaniu od klienta pakietu z zawartością: „REPLY” proces zakończy pracę.

Szczegóły implementacyjne:

- Serwer potrafi obsługiwać naraz dowolną liczbę klientów.
- Wątek sprawdzania poprawności sekwencji działa cyklicznie w pętli.
- Przy starcie aplikacji serwera następuje weryfikacja argumentów którymi został uruchomiony, jeżeli liczba portów jest niepoprawna program odpowiednio to zakomunikuje i zakończy pracę. Jeżeli typ danych portów podanych w argumentach nie będzie się zgadzał w dalszej części pracy programu, program się zakończy i wypisze odpowiednią wiadomość.

Klient

1) Start

Wątek klienta uruchamiamy poprzez wywołanie metody `main()` zawartą w klasie `UDPCClient`, która jako pierwszy argument przyjmuje IP klienta do którego ma wysłać pakiety UDP w celu autoryzacji.

2) Wysyłanie pakietów

Po uruchomieniu program zacznie wątek który jest odpowiedzialny za wysyłanie pakietów na zadanych w argumentach programu IP. Zawartością pakietów są kolejne litery alfabetu przy pierwszej jest zawsze losowa. Po wysłaniu pakietu wątek rozpocznie następny wątek który będzie odliczał do 30 sekund i jeżeli w tym czasie wątek klienta nie otrzyma pakietu z numerem portu zawartym w nim, to skończy prace. Razem z rozpoczęciem wątku, który odlicza do timeout'u programu, główny wątek oczekuje na pakiet z portem od serwera, jeżeli ten pakiet otrzyma rozpocznie połączenie TCP na podanym porcie z serwerem i po otrzymaniu pakietu z „REQUEST” wyśle pakiet z zawartością „REPLY” i zakończy pomyślnie swoją prace.

3) Połączenie TCP

Razem z rozpoczęciem wątku, który odlicza do timeout'u programu, główny wątek oczekuje na pakiet z portem od serwera, jeżeli ten pakiet otrzyma rozpocznie połączenie TCP na podanym porcie z serwerem i po otrzymaniu pakietu z „REQUEST” wyśle pakiet z zawartością „REPLY” i zakończy pomyślnie swoją prace.

Szczegóły implementacyjne:

- Przy starcie programu klienta następuje weryfikacja argumentów którymi został uruchomiony, jeżeli nie ma podanego IP albo portu program wypisze odpowiednia sentencje i zakończy prace. Jeżeli IP jest podane w złej formie również wypisze odpowiednie zdanie i zakończy prace.

3. TESTOWANIE

- w celu przetestowania czy serwer obsługuje jednego klienta naraz, uruchomiłem proces serwera i klienta, po czym obserwowałem komunikaty pojawiające się w konsoli.
- w celu przetestowania czy pakiety przychodzą od wielu klientów, uruchomiłem proces serwera i 2 klientów na różnych komputerach i obserwowałem informacje napisane w konsoli przez serwer.
- w celu przetestowania czy pakiety zapisały się w odpowiednich miejscach kolekcji, uruchomiłem proces serwera i 2 klientów i obserwowałem zawartości wypisywanych kolekcji na konsoli.

- w celu przetestowania czy autoryzacja zachodzi dla wielu klientów, uruchomiłem proces serwera oraz 2 klientów i obserwowałem konsolę czekając na pakiet z zawartością „REPLY” w danym kliencie.