

Zadanie 2: UDP port knocking SKJ (2018)

Wstęp

Zadanie polega na napisaniu pary aplikacji - serwera i klienta realizujących autoryzację metodą “UDP port knocking” (https://pl.wikipedia.org/wiki/Port_knocking).

Sposób działania aplikacji

Aplikacja składa się z dwóch procesów, serwera i klienta.

- W pierwszym rzędzie należy uruchomić proces serwera. Proces ten otwiera zadaną parametrem liczbę portów UDP, a następnie zaczyna na nich nasłuchować na pakiety od klientów. Jeśli zostanie wykryta odpowiednia sekwencja pakietów UDP wysyłanych z tego jednego adresu (ten sam adres IP i port), otwiera losowo wybrany port TCP i na adres z którego te pakiety przychodziły, wysyła komunikat UDP z numerem tego portu TCP. Następnie oczekuje połączenia TCP i po zakończeniu prostej komunikacji (typu zapytanie odpowiedź) rozłącza się, po czym ponownie zaczyna nasłuchiwać na portach UDP.
- Aplikacja klienta przyjmuje jako parametry adres serwera i numery portów, na które ma “pukać”. Po uruchomieniu otwiera port UDP, z którego wysyła serię pakietów na kolejne porty podane jako parametry. Po ich wysłaniu oczekuje na komunikat zwrotny zawierający numer portu TCP, z którym to następnie nawiązuje połączenie. Po wykonaniu prostej komunikacji z serwerem (typu zapytanie-odpowiedź) rozłącza się i kończy pracę.

Jeśli sekwencja pakietów UDP jest niepoprawna, odpowiedź od serwera nie przyjdzie. W takim przypadku, po określonym czasie (timeout) klient kończy pracę z komunikatem błędu.

Wymagania i sposób oceny

W celu realizacji zadania należy zaprojektować i zaimplementować procesy klienta i serwera implementujące własny protokół komunikacyjny, umożliwiający realizację opisanej powyżej

funkcjonalności. Kwestię tego, jakiej treści pakiety są przesyłane między poszczególnymi procesami pozostawia się autorowi.

1. Proces serwera uruchamiany jest parametrami będącymi listą numerów portów UDP, na ma oczekiwać na kolejne pakiety. Lista może być dowolnej długości i porty na niej mogą się powtarzać (oczywiście port otwieramy tylko raz). Zakładamy, że używane są jedynie porty o numerach powyżej 1024. Jeśli serwer nie może otworzyć danego portu z powodu jego zajętości przez inny proces - kończy pracę z błędem.
2. Proces klienta jako pierwszy swój parametr otrzymuje adres IP serwera, kolejnymi parametrami są numery portów UDP, na które należy wysyłać kolejne pakiety. Do pracy powinien używać losowych (ustalanych przez system) portów UDP i TCP.
3. Poprawny i pełny projekt wart jest **4 punkty**. Za zrealizowanie każdej z poniższych funkcjonalności można otrzymać punkty do podanej wartości:
 - **maksymalnie 2 punkty** za uzyskanie powyższej funkcjonalności przy założeniu tylko jednego klienta próbującego uzyskać autoryzację w danym momencie,
 - **maksymalnie 4 punkty** za uzyskanie powyższej funkcjonalności przy założeniu potencjalnie dowolnej liczby klientów próbujących uzyskać autoryzację w danym momencie.
4. Aplikację piszemy w języku Java zgodnie ze standardem Java 8 (JDK 1.8). Do komunikacji przez sieć można wykorzystać jedynie podstawowe klasy do komunikacji z wykorzystaniem protokołu UDP.
5. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX w nieprzekraczalnym terminie 30.12.2018 (termin może zostać zmieniony przez prowadzącego grupę).
6. Spakowany plik projektu powinien obejmować:
 - Plik *Dokumentacja(nr.indeksu)Zad2.pdf*, opisujący, co zostało zrealizowane, co się nie udało, jak zainstalować, gdzie ewentualnie są błędy, których nie udało się poprawić. W szczególności, plik musi zawierać szczegółowy opis zaprojektowanego i zaimplementowanego protokołu (brak opisu protokołu lub jego fragmentaryczność może spowodować znaczące obniżenie oceny rozwiązania zadania).
 - Pliki źródłowe (dla JDK 1.8) (włącznie z wszelkimi bibliotekami nie należącymi do standardowej instalacji Javy, których autor użył) - aplikacja musi dać się bez problemu skompilować na komputerach w laboratorium w PJA.

UWAGA: PLIK Z DOKUMENTACJĄ JEST WARUNKIEM KONIECZNYM PRZYJĘCIA PROJEKTU DO OCENY.

7. Prowadzący oceniać będą w pierwszym rzędzie poprawność działania programu i zgodność ze specyfikacją, ale na ocenę wpływać będzie także zgodność wytworzonego oprogramowania z zasadami inżynierii oprogramowania i jakość implementacji.
8. JEŚLI NIE WYSZCZEGÓLNIŁO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŻBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.