Projekt numer 3 z przedmiotu SKJ – semestr zimowy 2015/16

Temat: **Zdalnie konfigurowalny serwer proxy HTTP**

1. Zadanie polega na napisaniu aplikacji działającej jako serwer proxy HTTP, umożliwiającej jej dodatkową konfigurację zdalną w trakcie pracy, plus opcjonalnego programu konfiguratora.
2. Aplikacja składa się z jednego procesu pełniącego rolę serwera. Serwer pracuje na zadanym jako parametr porcie TCP i nasłuchuje na nim na połączenia od klientów – przeglądarek WWW. W celu wykorzystania takiego serwera jako proxy, przeglądarkę należy skonfigurować podając w jej konfiguracji adres serwera oraz numer portu (zgodnie z konfiguracją wybranej przeglądarki). Proxy HTTP działa na zasadzie przekazywania żądań – przeglądarka wysyła żądanie ściągnięcia danych nie bezpośrednio do serwera HTTP, a jedynie przekazuje je do pośrednika proxy. Pośrednik łączy się z serwerem występując jako strona komunikacji HTTP, a uzyskane wyniki przekazuje do klienta (przeglądarki). Należy pamiętać przy tym o konieczności właściwego przekazania do serwera HTTP wszystkich elementów zapytania HTTP wygenerowanych przez przeglądarkę.
    a. Implementacja samej funkcjonalności proxy HTTP może zostać zrealizowana samodzielnie, lub też można skorzystać z gotowej, znalezionej w sieci implementacji. Aczkolwiek własnoręczne wykonanie tej części podniesie rangę rozwiązania.
    b. Najważniejszym elementem pracy jest wzbogacenie samego mechanizmu proxy o mechanizm kontroli dostępu, sprowadzający się do możliwości dynamicznego blokowania dostępu do określonych treści oraz dla określonych klientów. W tym celu proxy udostępnia możliwość definiowania wielu list (ACL, Access Control List): list dozwolonych stron, blokowanych, list adresów hostów (klientów) oraz łączenia ich według schematów:
        i. klienci z listy X mają pozwolenie na dostęp do stron z listy A,
        ii. klienci z listy Y mają nie mogą oglądać stron z listy B.

        Można zdefiniować wiele list adresów oraz wiele list stron. Dany obiekt może należeć do wielu list jednocześnie.

        Można zdefiniować politykę domyślną:
        i. blokuj/deny (wszystko co niezdefiniowane jest zablokowane, dozwolone są jedynie dostępy zgodne ze zdefiniowanymi listami),
        ii. zezwalaj/allow (wszystko co niezdefiniowane jest dozwolone, zablokowane są jedynie dostępy zgodnie ze zdefiniowanymi listami).

        W zależności od aktualnie wybranej polityki domyślnej, przy każdym żądaniu rozpatrywane są odpowiednie listy i jeśli żądanie od zadanego klienta może zostać spełnione, jest oko wykonywane, w p.p. odpowiedzią powinien być komunikat odmowy dostępu (właściwy kod HTTP).
    c. Początkowo proxy startuje bez zdefiniowanych list, z domyślą polityką „zezwalaj”. Podczas pracy można połączyć się z proxy przez określony, z góry znany port za pomocą dodatkowego oprogramowania (może to być zwykły klient telnet dla komunikacji TCP lub własny „konfigurator” dla komunikacji UDP). Podczas połączenia po poprawnym uwierzytelnieniu za pomocą hasła (login niewymagany), za pomocą opracowanego protokołu można wybierać politykę domyślną, tworzyć nowe listy adresów, listy stron, wyświetlać ich aktualny stan, przypisywać do nich nową zawartość oraz usuwać obecną oraz łączyć zgodnie ze schematami podanymi powyżej.
    d. Tworzona na bieżąco konfiguracja powinna być zapisywana w pliku, który jest automatycznie wczytywany podczas startu proxy.
3. Protokół konfiguracyjny:
   W związku z wymogiem konfiguracji zdalnej, należy opracować i zaimplementować protokół konfiguracyjny pozwalający na dokonywanie operacji opisanych w punkcie 2c. Poprawny i pełny opis tego protokołu stanowi główną część zadania i będzie szczególnie dokładnie oceniony. Protokół ma pozwalać na:
    a. Wybór polityki domyślnej (allow/deny).
    b. Utworzenie nowej listy stron o określonej nazwie. Można zdefiniować wiele list stron.
    c. Wyświetlenie dostępnych list stron.

d. Wyświetlenie zawartości konkretnej listy stron.
e. Dodanie adresu URL do konkretnej listy stron. Przy dopasowywaniu zakładamy, że następuje ono na zasadzie prefiksu adresu.
f. Usunięcie konkretnej pozycji z konkretnej listy stron.
g. Usunięcie konkretnej listy stron.
h. Utworzenie listy adresów klientów o określonej nazwie. Można zdefiniować wiele list klientów.
i. Wyświetlenie dostępnych list klientów.
j. Wyświetlenie zawartości konkretnej listy klientów.
k. Dodanie adresu klienta do listy adresów. Dopuszcza się adresy IP jak i schematy adresowania postaci adres IP/maska (jak w przypadku adresowania podsieci IP).
l. Usunięcie konkretnego adresu klienta z konkretnej listy adresów.
m. Usunięcie konkretnej listy klientów.
n. Kojarzenia listy adresów stron z listą adresów klientów według schematów podanych w punkcie 2b.

Protokół powinien być zwykłym protokołem tekstowym, w którym polecenie wygląda na przykład tak (poniższe to jedynie fragmentaryczna propozycja):

```
DEFAULT DENY
CREATE ADDRESS LIST listaAdresów
ADD URL url TO listaAdresów
CREATE CLIENT LIST listaKlientów
ADD CLIENT klient TO listaKlientów
ALLOW listaKlientów FOR listaAdresów
```

W wyniku powyższej sekwencji domyślną polityką stanie się blokowanie (DENY), zostaną utworzone dwie listy, jedna adresowa, druga dla klientów, do każdej zostanie dodany jeden wpis, a następnie zadany host dostanie pozwolenie na komunikację zadaną stroną HTTP.

Do komunikacji z proxy w celu komunikacji należy użyć zwykłych gniazd TCP (klasy ServerSocket i Socket). Programem klienckim może być w takim przypadku dowolny klient telnet.

4. Konfigurator:
   Dodatkowe punkty można uzyskać za **dodatkową** implementację konfiguracji za pomocą protokołu UDP. W tym celu należy napisać własną aplikację, która będzie pośredniczyć w komunikacji między użytkownikiem a serwerem. Wymaga to również uzupełnienia kodu serwera. Proszę pamiętać o wymogu autoryzacji. Do komunikacji należy używać wyłącznie prostych gniazd UDP (klasa DatagramSocket).
5. Kwestię przechowywania wpisów, identyfikatorów itp. pozostawia się autorowi. Bieżąca konfiguracja serwera musi być dostępna po jego restarcie (musi być składowana na dysku).
6. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX do ??.??.???? (termin może zostać zmieniony przez prowadzącego daną grupę).
7. Za poprawne rozwiązanie tego zadania można uzyskać do 6 punktów (plus 1 dodatkowy):
   a. 3 punkty za pełną implementację funkcjonalności opisanych w punktach 2 i 3.
   b. 3 punkty za pełny i poprawny opis zaimplementowanego protokołu konfiguracyjnego (punkt 3).
   c. Dodatkowo można uzyskać 1 punkt za implementację programu konfiguratora komunikującego się z serwerem proxy za pomocą protokołu UDP (punkt 4).
8. Spakowane archiwum z plikami projektu powinno zawierać:
   a. Pliki źródłowe (dla JDK 1.8),
   b. Pliki binarne (class),
   c. Skrypty startowe, umożliwiające uruchomienie implementacji (opcjonalne),
   d. Plik Readme.txt z opisem i uwagami autora, w szczególności:
      i. **<u>szczegółowy opis zaprojektowanego i zaimplementowanego protokołu</u>** (brak opisu protokołu lub jego fragmentaryczność może spowodować znaczące obniżenie oceny rozwiązania zadania),

   ii. jak zainstalować,
   iii. jak uruchomić to co działa,
   iv. jak używać,
   v. co nie działa (jeśli nie działa).
9. JEŚLI NIE WYSZCZEGÓLNIONO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŹBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.