Dokumentácia k projektu:

Filtrující DNS resolver

Samuel Križan (xkriza06)
11.11.2020

Obsah

1 Úvod

Zadaním bolo vytvoriť komunikujúcu aplikáciu pomocov knižnice BSD sockets v jazyku C/C++. Vybral som si variantu filtrujúceho DNS resolvera. Ten mal fungovať na princípe DNS servera, ktorý prijme DNS dotaz a preženie ho cez filter blokovaných adries. V prípade, že je daná adresa medzi blokovanými, pošle sa späť chybová správa. V opačnom prípade bude dotaz preposlaný reálnemu resolveru a jeho odpoveď bude následne preposlaná inicializátorovi dotazu.

1.1 Obmedzenia

- Protokoly: IPv4, UDP, DNS,
- typy dotazov: A,
- počet dotazov vrámci 1 packetu: 1.

1.2 Spustenie a použitie

./dns -s *server* [-p *port*] -f *filter_file*

- *server* = ip adresa/doménové meno DNS servera, kam sa majú dotazy preposielať,
- *port* = číslo portu, na ktorom sa budú zachytávať dotazy, predvolené je 53,
- *filter_file* = názov súboru, v ktorom sú blokované domény,
- argumenty môžu byť v ľubovolnom poradí.

2 Návrh

Prvé zo všetkého je nutné spracovať vstupné argumenty a následne načítať informácie zo zadaného súboru. Potom je možné prejsť na samotné jadro programu, čím bude klasický UDP server, ktorý bude príjmať DNS sokety na zadanom porte. V prijatom sokete nájde dotazovaný názov a jeho typ. Ak je daný typ nie je podporovaný, nastaví príznaky na odpoveď a neimplementovanú operáciu a odošle ho späť, inak pokračuje ďalej. Získaný názov porovná s názvami načítanými zo súboru. Ak nájde zhodu, v danom sokete nastaví príznaky odpovede a odmietnutého dotazu a pošle ho späť odosielateľovi. Ak nenájde zhodu, server začne vystupovať ako DNS klient a soket pošle na požadovaný DNS server, od ktorého následne príjme odpoveď na dotaz. Túto odpoveď potom prepošle znova ako server späť pôvodnému odosielateľovi.

3 implementácia

Program je tvorený viacerými zdrojovými súbormi, ktoré tvoria logické celky a mali by tak uľahčiť pochopenie danej logiky.

3.1 Spracovanie argumentov a načítanie súboru

- params.cpp, loadFile.cpp

Na spracovanie argumentov využívam vlastnú triedu *Param*, ktorá ich na základe jedonduchého stavového automatu roztriedi a uchová vo vhodnej forme. Súbor načítavam klasicky, prechádzaním riadok po riadku a jednotlivé záznamy ukladám vo forme std::vector <string>.

3.2 Server

- server.cpp

Funkcia *server_run()* tvorí celú podstatu programu. Vytvorí UDP server na zadanom porte a v nekonečnom cykle bude prijímať sockety. Aby sa so socketom lepšie pracovalo, po prijatí ho namapuje do štruktúry[1] hlavičky DNS správy. Následne z neho získa doménové meno[2] a typ dotazu[3]. Overí sa, či je daný typ podporovaný:

a) Ak je, pokračuje sa ďalej,

b) ak nie je, v sockete sa nastaví QR bit na 1(response) a RCODE sa nastaví na 4(not implemented). Následne sa socket pošle späť odosielateľovi.

Potom iterovaním cez std:vector ho porovná so záznamami zo súboru.

a) Ak bola nájdená zhoda, v sockete sa nastaví QR bit na 1(response) a RCODE sa nastaví na 5(refused). Následne sa socket pošle späť odosielateľovi,

b) ak nebola nájdená zhoda, zpreloží sa ip adresa DNS resolvera pomocou *gethostbyname()* a zavolá sa *funkcia client_run()*, ktorá vracia DNS odpoveď od resolvera. Tá sa nezmenená pošle odosielateľovi.

3.3 Client

- client.cpp

Funkcia *client_run()* sa správa ako udp client. Na prebratú adresu pošle prebratú správu a príjme odpoveď. Túto odpoveď v nezmenenom tvare vráti v návratovej štruktúre spolu s jej veľkosťou.

1 Štruktúra DNS_PCK v súbore head.hpp, inšpirácia z sourcedaddy.com
2 Za hlavičkou pevnej dĺžky je uložené doménové meno skladajúce sa zo sekvencie čísiel a znakov, kde číslo udáva počet nasledujúcich znakov, ktoré tvoria meno. Táto sekvencia je ukončená prázdnym znakom.
3 Typ sa nachádza na 2 bytoch za prázdnym znakom.

3.4 Chybové stavy

Program sa ukončí chybovým stavom len v 3 prípadoch:

a) Sú zle zadané vstupné argumenty,

b) nepodarilo sa načítať dáta zo vstupného súboru,

c) nepodarilo sa vytvoriť socket pre serverovú časť.

Po všetkých ostatných chybových stavoch je program schopný pokračovať ďalej, s tým že na štandardný chybový výstup sa vypíše hláška oznamujúca danú chybu.

Zdroje

- [RFC1035]
- https://sourcedaddy.com/networking/dns-protocol.html