VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technológií


Síťové aplikace a správa sítí 2020

# Manuál k programu
# Filtrujúci DNS Resolver

Damián Sova(xsovad06)

Senica, 18.11.2020

# ÚVOD

Znenie zadania programu je nasledovné: „Napište program dns, který bude filtrovat dotazy typu A směřující na domény v rámci dodaného seznamu a jejich poddomény. Ostatní dotazy bude přeposílat v nezměněné podobě specifikovanému resolveru. Odpovědi na dříve přeposlané dotazy bude program předávat původnímu tazateli. Analýza a sestavení DNS zpráv musí být implementována přímo v programu dns. Stačí uvažovat pouze komunikaci pomocí UDP a dotazy typu A. Na jiné typu dotazů a nežádoucí dotazy odpovídejte vhodnou chybovou zprávou.“

# DNS

Domain name systém (DNS), je systém, ktorý ukladá prístup k informácií o názve stroja a názve domény v databáze. Najdôležitejšie je, že poskytuje mechanizmus získania IP adresy pre každé meno stroja a naopak. DNS poskytuje dôležitú službu, pretože kým počítače a sieťový hardware pracujú s IP adresami, ľudia si ľahšie pamätajú doménové mená pri ich používaní. DNS je medzičlánok medzi človekom a strojom.

# DNS packet

DNS packet sa skladá z nasledujúcich častí:

- Header (hlavička)
- Question (otázka)
- Answer (odpoveď)
- Authority (autorizovaná odpoveď)
- Additional (naviac odpovede)

# Header

Hlavička s pevnou veľkosťou je súčasťou každého DNS packetu, má velkosť 12B a skladá sa z nasledujúcich častí:

- |ID| - identifikačné číslo packetu (2B)
- |QR| - flag identifikujúci, či sa jedná o otázku alebo odpoveď (1b)
- |OPCODE| - označuje variantu balíku (4b)
- |TC| - flag identifikujúci poškodený balík (1b)
- |RD| - flag identifikujúci či je vyžiadaná rekurzia (1b)
- |Z| - rezerované miesto (1b)
- |RA| - flag identifikujúci či je server dokáže vykonať rekurziu (1b)
- |QDCOUNT| - číslo identifikujúce počet otázok (2B)
- |ANCOUNT| - číslo identifikujúce počet odpovedí (2B)
- |NSCOUNT| - číslo identifikujúce počet autorizovaných odpovedí (2B)
- |ARCOUNT| - číslo identifikujúce počet naviac odpovedí (2B)

# Question

Táto časť ma premennú veľkosť a je súčasťou každého DNS packetu.

Skladá sa z nasledovných častí:

- |CNAME| - meno domény, ktorá má byť preložená
- |QTYPE| - typ záznamu
- |QCLASS| - trieda komunikácie

V tomto projekte potrebujeme použiť len doménové meno pre filtrovanie nežiadúcich domén a typ záznamu, kde podporujeme len záznamy typu „A“.

# CNAME

Meno domény musí byť podľa normy rozdelené ".", na štítky. Pred každý štítok je pridané číslo, označujúce počet znakov štítku.

# QTYPE

Typ záznamu, ktorých je mnoho, pre naše účely si vystačíme len s jediným:

- |A| - záznam obsahujúci IPv4 adresu

Ostatné časti DNS packetu nás nezaujímajú a nekontrolujeme ani nemeníme ich obsah, preto ich nebudeme bližšie opisovať.

# ODOVZDANÉ SÚBORY

Odovzdaný je jeden súbor xsovad06.tar, ktorý obsahuje:

## dns.cpp

Súbor s funkčným zdrojovým kódom, implementáciou jednotlivých pomocných funkcií a hlavnej funkcie main().

## Header.hpp

Hlavičkový súbor obsahujúci importované knižnice, globálne premenné, deklaráciu pomocných štruktúr, deklaráciu triedy Arguments a prototypy pomocných funkcií.

## Makefile

Súbor obsahujúci príkaz na kompiláciu a príkaz na vymazanie súboru s príponou .o

## README.md

Súbor obsahuje základné informácie o programe, jeho funkčosti, používaní a zoznam odovzdaných súborov.

# NÁVRH RIEŠENIA

Prvou vecou je sparsovanie programových argumentov, z ktorých získame informácie, kto je náš DNS server, kde budeme zasielať dotazy od užívateľa. Ďalej máme priložený súbor s nežiadúcimi doménamy a poprípade číslo portu, na ktorom máme počúvať. Riešenie spočíva v týchto bodoch:

1. Otvoríme komunikačný kanál s klientom, kde budeme zachatávat DNS dotazy.
2. Odchytíme packet a z jeho hlavičky skontrolujeme typ záznamu, ak nie je typu „A“, upravíme Flagy v hlavičke a pošleme ho späť klientovi, inak pokračujeme.
3. Prečítame doménové meno, ktoré je uložené v časti packetu „Question“ a porovnáme ho s doménami v priloženom súbore. Ak nájdeme zhodu upravíme Flagy v hlavičke a pošleme packet späť klientovi, inak pokračujeme.
4. Otvoríme komunikčný kanál s DNS serverom a pošleme mu daný dotaz od klienta a obdržanú odpoveď prepošleme klientovi a obsluhujeme ďalšie dotazy.

# IMPLEMENTÁCIA

Pre účel implementácie bol z možných programovacích jazykov vybraný C++. Program nie je objektovo-orientovaný, i keď je použitá jedna trieda, ktorá reprezentuje prácu so spracovaním argumentov príkazovej riadky. Pri implementácii sa postupovalo tak, ako je spomenuté v návrhu riešenia.

Prvá vec programu, ktorá sa vykoná je spracovanie argumentov. To spracováva trieda Arguments prostredníctvom funkcie Arguments* arg_processor(int argc, char **argv). Tá preberá počet argumentov a pole týchto argumentov. V jej tele prostredníctvom funkcie getopt_long spracuje argumenty, ktoré triedi pomocou „switchu“. Pri spracovaní mena súboru pomocou funkcie int load_file(Arguments *arguments, std::string filter_file) otvorí daný subor a načíta ho do vektoru stringov undesired_domians, ktorý je objektová premenná.

Ďalej vytvoríme komunikačný kanál s klientom pomocou funkcie int bind_to_user(string port_number), ktorá vytvorí socket funkcou socket() pre komunikáciu a napojí sa na port(predvolený 53, inak podľa argumentu z príkazovej riadky).

Teraz prichádza na rad nekonečný cyklus, ktorý slúži na nekonečnú obsluhu DNS dotazov od klienta. V tomto cykle púšťame 3 funkcie, ktoré tvoria obsluhu všetkých požiadavkov a sú to nasledovné funckie:

- int user_query_handler(Arguments *args);
- int create_dns_connection(const char *server);
- int dns_query_user_response();

Funkcie vracajú „integer“ ako návratovú hodnotu, možnosti návratových kódov:

- EXIT_SUCCESS -hodnota 0
- EXIT_FAILURE -hodnota -1
- END -hodnota -2(Prišlo k odoslaniu odpovede klientovi vo funkcii)

### *user_query_handler(Arguments *args)*

Funkcia príjme dotaz od klienta pomocou funkcie revcfrom() a uloží ho do štruktúry dns_header, ktorú sme si zadefinovali v hlavičkovom

súbore. Potom posunutím odkazu na obdržané dáta o veľkosť štruktúry dns_header získame pointer na položku, z časti packetu „Question“, cname. Získaný pointer pošleme do funkcie char* cname_handler(char *start_label), ktorá načíta doménové meno do globálnej premennej target_domain a vráti pointer na zvyšné položky časti „Question“, ktoré si uložíme do štruktúry dns_question. Skontrolujeme QR flag z hlavičky, ktorý hovorí, či ide o otázku(query) alebo odpoved(response), ak je to otázka kontolujeme položku qtype zo štruktúry dns_question, ktorá musí byť typu „A“. Ak nie sú splnené požiadavky, nastavíme RCODE na hodnotu 4(NOTIMP) a pošleme packet späť. Inak pokračuje porovnávaním target_domain s undesired_domains a ak nájde zhodu, nastaví RCODE na hodnotu 5(REFUSED) a posiela paket späť.

### *int create_dns_connection(const char *server);*

Pomocou funkcie getaddrinfo() zistíme potrebné informácie na vytvorenie komunikačného kanálu s DNS serverom, ktorý bol určený cez príkazovú riadku. Získané informácie použijeme na vytvorenie socketu funkciou socket() a naviežeme spojenie funkcoiu connect().

### *int dns_query_user_response();*

Do vytvoreného kanálu posielame pôvodný packet od klienta funkciou send(). Nastavíme timeout pre odpoveď cez funkciu setsockopt() a čakáme na odpoveď. Tú prijmeme použitím recv() a následne obdržanú odpoveď prepošleme klientovy na prvý komunikačný kanál cez sendto().

# IMPLEMENTÁCIA POMOCNÝCH FUNKCIÍ

### *void print_help()*

Vypíše pomocnú správu, ak je zadaný samostatný argument programu --help/-h. Prípadne ak boli nesprávne zadané argumenty.

### *void error_message(const char *message)*

Vypíše chybovú hlášku na štandardný chybový výstup stderr a ukončí program s chybovým kódom EXIT_FAILURE.

# PRÍKLAD SPUSTENIA

Program preložíte príkazom **make**. Pre prípadne odstánenie „*.o“ súborov použijeme príkaz **make clean**.Pre spustenie použijete príkaz

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

Možnosti argumentov:

- -s <IP_adresa|Doménové_meno> Server pre dnsdotazy
- -p <Číslo_portu> Komunikácia s klientom na porte(nepovinný argument)
- -f <Filter_súbor> Súbor na obsahujúci nechcené domény

Ďalej je možné použiť nekombinovateľný argument na vypísanie pomocnej správy. Pri kombinovaní sa program ukončí chybou. Použitie:

- -h alebo –help

# REFERENICIE

https://tools.ietf.org/html/rfc1035

https://sourcedaddy.com/networking/dns-protocol.html

https://www2.cs.duke.edu/courses/fall16/compsci356/DNS/DNS-primer.pdf

https://man7.org/linux/man-pages/man3/getaddrinfo.3.html

https://stackoverflow.com/

https://www.google.com/