

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí
Filtrující DNS resolver

Obsah

1	Problematika	2
1.1	Úvod	2
1.2	Komunikácia; Protokol DNS	2
1.3	DNS paket	2
2	Návrh aplikácie a implementácia	3
2.1	Priebeh	3
2.2	Riešenia podproblémov	4
3	Použitie	4
3.1	Server	4
3.2	Klient	5
4	Referenčná literatúra	6

1 Problematika

1.1 Úvod

Systém doménových mien, skr. DNS, je systém, ktorý ukladá prístup k informácii o názve stroja a ďalších identifikátorov sieťových zariadení a služieb v istej databáze. Systém poskytuje mechanizmus získania IP adresy pre každé meno stroja a naopak, a uvádza poštové servery akceptujúce poštu pre danú doménu. Zmyslom DNS je uľahčiť prácu na internete ľuďom jednoduchším zapamätaním si mien strojov a domén v čitateľnej forme než počítačmi používajúce IP adresy. DNS teda tvorí prostredníka medzi človekom a softvérom. 1

1.2 Komunikácia; Protokol DNS

Komunikácia prebieha nad UDP, na porte 53.

1.3 DNS paket

Analýza paketu prebieha do hĺbky hlavičky a otázky.

Formát DNS paketu sa skladá z 5 častí:

- Header
- Question
- Answer
- Authority
- Additional

Formát hlavičky (Header) sa skladá z 13 častí:

- ID
- QR
- Opcode
- AA
- TC
- RD
- RA
- Z
- RCODE
- QDCOUNT
- ANCOUNT
- NSCOUNT
- ARCOUNT

Formát otázky sa skladá z 5 častí:

- QNAME
- QTYPE
- QCLASS

2 Návrh aplikácie a implementácia

Program bude filtrovať požiadavky typu A smerujúce na domény v rámci dodaného zoznamu a ich poddomény. Ostatné požiadavky v nezmenenej forme prepošle špecifikovanému resolveru. Odpovede na predchádzajúce požiadavky prepošle pôvodnému klientovi.

Riešenie je obsiahnuté v jednom súbore; `dns.cpp`.

Obsahuje:

- **Štruktúry**
 - `dns_header` : Zložky podľa podľa formátu hlavičky
 - `Args` : Zadané argumenty a flagy
- **Funkcie**
 - `getAtgumets` : Sparsuje a zhodnotí argumenty
 - `err` : Vypíše chybovú hlášku a ukončí program
 - `parse_filter` : Sparsuje zadaný filter

2.1 Priebeh

Program postupuje nasledovne:

1. Program na začiatku zavolá funkciu `getAtgumets` na uloženie parametrov do `Args`
2. Získa informácie od špecifikovanom DNS serveri
3. Vytvorí IPv6 socket pre zadaný server
4. Načíta informácie o zadanom filtri do vektoru
5. Vytvorí lokálny IPv6 socket a nastaví ho na prijímanie dotazov
6. V tomto bode bude program počúvať a očakávať žiadosti
7. Po príchode dotazu skontroluje jeho `QR`, `OPCODE` v hlavičke a získa požadovanú doménu
8. Kontrola prítomnosti domény vo filtri
 - (a) **Prítomná**: pošle odpoveď klientovi so statusom "REFUSED" a vráti sa na bod 6.
 - (b) **Neprítomná**: pokračuje
9. Prepošle prijatý dotaz na špecifikovaný DNS server a čaká na jeho odpoveď
 - (a) **Pošle odpoveď**: prepošle ju klientovi a vráti sa na bod 6
 - (b) **Timeout**: pošle odpoveď klientovi so statusom "SERVFAIL" a vráti sa na bod 6.

2.2 Riešenia podproblémov

Kvôli možnému zadaniu DNS servera v IPv6 formáte sú sockety vytvorené ako IPv6 sockety s možnosťou zachytávania IPv4 komunikácie.

Filtrácia prebieha najprv získaním stringu domény z paketu, jeho konverzii na štandardné doménové meno, následne sa hľadá string z vektora filtrov či je obsiahnutý v názve domény. V kladnom prípade nastane ešte kontrola či je to subdoména, presne rovnaká doména alebo úplne iná doména náhodne obsahujúca pasáž z filtrovanej domény.

V prípade zadania invalidného DNS servera je nastavený Timeout na 5 sekúnd pri čakaní na odpoveď.

3 Použitie

Program sa preloží príkazom *make* v zložke so súbormi *Makefile* a *dns.cpp*. Spustenie na strane serveru s parametrom *verbose*:

3.1 Server

```
$ ./dns -s ::ffff:8.8.8.8 -p 5454 -f filter.txt -v
# Opening a local UDP socket
# Binding to the port 5454 (19989)

# Waiting on port 5454
# Request received from port 43295, bytes 53
Forwarding Packet...
Receiving Packet...
Sending Answer...
Done...

# Waiting on port 5454
```

3.2 Klient

```
$ dig @localhost -p 5454 google.com
```

```
; <<>> DiG 9.16.1-Ubuntu <<>> @localhost -p 5454 google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3994
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: Message has 457 extra bytes at end

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                 187     IN      A      172.217.23.238

;; Query time: 84 msec
;; SERVER: 127.0.0.1#5454(127.0.0.1)
;; WHEN: St nov 18 20:29:56 CET 2020
;; MSG SIZE rcvd: 512
```

```
$ dig @localhost -p 5454 ok.co
```

```
; <<>> DiG 9.16.1-Ubuntu <<>> @localhost -p 5454 ok.co
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 36082
;; flags: qr rd ad; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available
;; WARNING: Message has 466 extra bytes at end

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5386e20d8a14b76c (echoed)
;; QUESTION SECTION:
;ok.co.                     IN      A

;; Query time: 20 msec
;; SERVER: 127.0.0.1#5454(127.0.0.1)
;; WHEN: St nov 18 20:30:28 CET 2020
;; MSG SIZE rcvd: 512
```

4 Referenčná literatúra

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

<https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FISA-IT%2Flectures%2Fisa-dns.pdf&cid=14020>

https://sk.wikipedia.org/wiki/Syst/%C3%A9m_n%C3%A1zvov_dom%C3%A9n

<https://linux.die.net/man/>

<http://www.cplusplus.com/reference/>

<https://www.geeksforgeeks.org/udp-server-client-implementation-c/>

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