Vysoké učení technické v Brně
Fakulta informačních technologií

Síťové aplikace a správa sítí –

Filtrující DNS resolver

16. listopadu 2020 Jakub Sekula

Obsah

1 DNS

DNS je hierarchický systém doménových jmen. Ten realizují právě DNS servery spolu s protokolem stejného jména, který využívají k výměně informací. DNS je v praxi překladová služba, která číselnou IP adresu přeloží do podoby zvolené domény. Na IP se ptá DNS serveru[1].

2 IPV4 a IPV6

Pro komunikaci mezi klientem a serverem je využit IPV6 soket. Většina operačních systémů totiž podporuje příjem ipv4 paketu ipv6 soketem. Tzn. pokud by server bežel na operačním systému, který toto nepodporuje, tak server nebude fungovat. Server používá 2 sokety. Jeden pro komunikaci mezi klientem a serverem a druhý používá pro komunikaci s DNS serverem. Zdali bude soket ipv4, nebo ipv6, rozhoduje parametr -s, respektive funkce getaddrinfo(), kterou volám pro zjištění ipv6 adresy serveru a pokud funkce nevrátí ipv6 adresu, tak volám znovu, ale tentokrát pro ipv4.

3 DNS packet

Veškerá komunikace v domain protokolu je přenášená pomocí zpráv, které mají následující podobu[3].

```
+---------------------+
|        Header       | hlavička
+---------------------+
|       Question      | dotaz
+---------------------+
|        Answer       | odpověd'
+---------------------+
|      Authority      | autorita( autorizovaná odpověd' )
+---------------------+
|      Additional     | dodatečné informace
+---------------------+
```

3.1 Hlavička

Hlavička paketu je přítomna vždy. Hlavička zahrnuje pole, které značí, z jakých dalších částí se paket skládá a dále také, jestli je paket dotaz, nebo odpověď a další[3]. Strukturu pro uchování hlavičky jsem převzal a lehce poupravil[2]. Podoba hlavičky je následující:

- ID - 16 bitový identifikátor paketu
- QR - 1 bit značící, jestli je paket typu 0 (dotaz), nebo 1 (odpověď)
- Opcode - 4 bity pro označení typu dotazu
- AA - 1bit Odpověď autority, tedy serveru odpovědného serveru
- TC - 1 bit Indikuje zda byl paket zmenšen

- RD - 1 bit Indikuje, zda-li je vyžadována rekurze
- RA - 1 bit Podpora rekurze
- Z - 4 bity Rezerva pro budoucí použití
- RCODE - 4 bity Response code
- QDCOUNT - 16 bitů Specifikujících počet vstupů v dotazu
- ANCOUNT - 16 bitů Specifikujících počet zdrojových záznamů v odpověďi
- NSCOUNT - 16 bitů Počet zdrojových záznamů
- ARCOUNT - 16 bitů Počet zdrojových záznamů v přídavné sekci

3.2 Dotaz

Část paketu používaná pro přenos dotazu a parametrů s ním spojených[3].

- QNAME - 16 bitů Název domény
- QTYPE - 16 bitů Typ dotazu
- QCLASS - 16 bitů Třída dotazu

3.3 Odpověď

Část paketu používaná pro přenos odpověďi a parametrů s ní spojených[3].

- NAME - 16 bitů Název domény spojené se zdrojovým záznamem
- TYPE - 16 bitů Typ dotazu
- CLASS - 16 bitů Specifikuje význam dat v RDATA
- TTL - 16 bitů Čas po jakou dobu má být zaznám cachován před vymazáním
- RDLENGTH - 16 bitů Délka oktetů v RDATA
- RDATA - Proměnná délka oktetových stringů popisujících zdroj

3.4 Authority

Sekce obsahuje zdrojové záznamy odkazující na autoritativní server[3].

3.5 Additional

Obsahuje zdrojové záznamy, které jsou spjaté s dotazem, ale nejsou odpovědí na tento dotaz[3].

3.6 Paket s vícero dotazy

V rfc pro dns paket[3] jsem se dočetl o tom, že v paketu by mohlo být vícero dotazů. Toto však ale v dns programu není podporováno.

4 Návrh a implementace aplikace

4.1 Překlad programu

V souboru se nachází Makefile, pomocí kterého lze pracovat s aplikací. Příkazem make se přeloží projekt a vygeneruje se program dns. Příkazem make clean se program dns smaže. Příkazem make test se spustí automatické testy. Makefile také zná příkaz make docu, který vytvoří doxygen dokumentaci k projektu.

4.2 Ukázka spouštění

Program se spouští následujícím způsobem:

./dns -s 1.1.1.1 -p 8080 -f tests/blacklist

./dns -s 8.8.4.4 -f tests/blacklist

4.3 Parametry programu

Program umí zpracovat 4 parametry. Jsou jimi:

- -s: IP adresa nebo doménové jméno DNS serveru (resolveru), kam se má zaslat dotaz. POZOR !! Pokud nebude zadán reálný DNS server, tak server bude posílat požadavky na parametrem zadaný server a je mu jedno, že se nejedná o DNS ! Tím pádem se klient nikdy nedočká odpovědi.
- -p port: Číslo portu, na kterém bude program očekávat dotazy. Výchozí je port 53. (volitený parametr)
- -f filter_file: Jméno souboru obsahující nežádoucí domény.
- -h help: Nápověda k programu

Parametry zpracovává funkce `getArguments()`

4.4 Tělo programu

Aplikace dns běží jako server. Po spuštění programu se vytvoří funkcí `socket()` komunikační soket, přiřadí se mu adresa pomocí funkce `bind()` a následně se v cyklu čeká na přijetí paketu na adrese localhost na portu zadaném parametrem programu `-p`.

Jakmile je zachycen paket na dané adrese a portu, tak se hlavní aplikační proces rozdělí funkcí `fork()`. Zatímco child proces získavá doménové jméno z paketu a další informace, hlavní proces se vrátí a čeká na přijetí dalšího paketu.

4.5 Child proces

Jakmile je paket přijat, zpracovává se v child procesu. V tomto procesu program získá dotazované doménové jméno a QTYPE z paketu, aby si ověřil, zda se opravdu jedná o dotaz typu A.

Dále program prochází zadaný soubor blacklistových domén a hledá, zda-li dotazovaná doména není na tomto seznamu nebo jestli není poddoménou na tomto seznamu.

Případnou zakázanou doménu vrací klientovi jako původní paket s nastaveným `rcode` flagem na 5 a flagem `qr` na 1. Obdobně je to pro případ jiného typu dotazu než je typ A, respektive s `rcode` nastaveným na 4.

V případě, že tyto kontroly proběhnou v pořádku, je paket přesměrován na DNS server zadaný parametrem programu `-s`. Parametr -s se předává funkci `getDnsIp()`, která vrací ip adresu DNS serveru. Po získání ip adresy je paket funkcí `sendto()` poslán na port 53 DNS serveru. Příchod odpověďi je očekáván funkcí `recvfrom()` a jakmile tento paket dorazí, je poslán nazpět klientovi.

4.6 Návratové kódy

- 9 - Chyba soubor neexistuje
- 10 - Chyba vstupních parametrů
- 11 - Chyba získávání ip adresy DNS serveru
- 12 - Chyba vytvoření soketu
- 13 - Chyba přidělování adresy soketu

4.7 Čísla portů

Program dns pracuje s čísly portu v rozmezí `<0, 65535>`. Při hodnotách mimo tento interval, program končí s chybou. Implicitně se bere port 53.

5 Testování

Korektní funkčnost aplikace jsem testoval pomocí nástrojů nslookup a dig. Testy fungovaly následovně. V jednom okně terminálu jsem pustil svůj program a ve 2. okně terminálu jsem pustil nslookup takto: `"nslookup -port=<port> -type=a doména localhost"` a takto dig: `"dig @localhost -p <port> <doména>"` respektive `"dig @ip6-localhost -p <port> <doména>"` a pak jsem porovnával ip adresu, která přišla zpět nslookupu a digu s online resolv nástrojem DNS Checker[1]. Případně jsem zadal ip adresu do prohlížeče. Takto jsem iteroval přes různé domény. Pro testování jsem také napsal skript pro python3 test.py. Skript se pouští příkazem make test. Skript byl napsán pro python 3.8.2. Jako povinné parametry skript má: číslo portu, soubor s doménami, soubor s blacklist doménami, ip adresa resolveru. Pro pomoc při spuštění skriptu stačí skript spustit bez parametrů a vypíše se nápověda.

5.1 Testování typu dotazu

Korektnost ošetření dns dotazu typu A jsem testoval tak, že jsem pustil nslookup bez parametru -type=a. Tím pádem se pošle první ipv4 paket a následně i ipv6 paket. V tomto případě nebyl QTYPE z hlavičky paketu nastaven na 1 ale na 28. V situaci kdy přišel ipv6 paket, dns program

[1]DNS Checker: https://dnschecker.org/.

reagoval korektně a poslal zpět paket s nastaveným `rcode` a `qr` a dále neposílal paket na dns resolver. Toto jsem kontroloval programem Wireshark, který celou komunikaci odchytával.

5.2 Testovaní blacklistových domén

Testování blacklistových domén probíhalo tak, že jsem si do souboru zapsal několik domén. Doménami byly kupříkladu facebook.com, docs.google.com a wis.fit.vutbr.cz a na bežící program dns jsem pomocí nslookup posílal domény, které jsem čekal, že projdou, tedy například face.com, google.com, google.cz, vutbr.cz, docs.google.cz apod. Následně jsem posílal domény, u kterých jsem čekal, že neprojdou, tedy domény jako google.com, wis.wis.fit.vutbr.cz atd. Celou síťovou komunikaci jsem sledoval programem Wireshark. A pro každou zaslanou doménu na server, jsem kontroloval odchozí paket na klienta a díval se na `rcode` a `qr` flagy v hlavičče.

5.3 Testovací prostředí

Program byl testován na linux mint 19 a serverech merlin a eva.

6 Blacklistové domény

Porovnávání zadaných domén s doménami ze souboru je case insensitive. Tzn. že pokud je v souboru zadán google.com a uživatel chce zjistit adresu serveru google.COM tak tato adresa neprojde.

Reference

[1] BEST-HOSTING s.r.o., i.-h.: Co je to DNS server? 2020. Dostupné z: https://best-hosting.cz/cs/napoveda/co-je-to-dns-server

[2] IoTh1nkN0t: DNS header for C. 2016. Dostupné z: https://0x00sec.org/t/dns-header-for-c/618

[3] Mockapetris, P.: Domain names - implementation and specification. Nov 1987: str. 1–55, doi:10.17487/rfc1035. Dostupné z: https://tools.ietf.org/html/rfc1035