VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí
2020/2021

Filtrující DNS resolver

17. listopadu 2020

xmusko00

Obsah

1 Zadání

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.

Při vytváření programu je povoleno použít pouze knihovny pro práci se sokety a další obvyklé funkce používané v síťovém prostředí (jako je netinet/*, sys/*, arpa/* apod.), knihovnu pro práci s vlákny (pthread), signály, časem, stejně jako standardní knihovnu jazyka C (varianty ISO/ANSI i POSIX), C++ a STL. Jiné knihovny nejsou povoleny.

Spuštění aplikace: dns -s server [-p port] -f filter_file

2 Úvod do problematiky

2.1 DNS [5]

Každé síťové rozhraní má danou IP adresu, která ho identifikuje. IP adresy jsou ale špatně zapamatovatelné, proto se používá doménové jméno, které je třeba ale pro účely komunikace přeložit. K tomu nám složí služba DNS (Domain Nam System), která právě mimo jiné je schopna zajišťovat překlad IP adres na doménové jméno a zpět.

Mezi DNS služby patří např.:

- Překlad doménových jmen na IP adresy
- Překlad IP adres na doménová jména
- Určení poštovního servru pro danou doménu
- Delegování zprávy domén na jednotlivé subjekty

Protokol DNS je používá porty TCP/53 i UDP/53 a je definován v RFC1035 [1]

DNS server

DNS server je aplikace, která si uchovává záznam doménový doménových jmen. Pokud pak přijde dotaz na DNS databázi, sever na něj odpoví.

Existují 3 typy DNS servrů:

- primární: obsahuje lokálně úplné záznamy doménách, které spravuje
- sekundární: data získává od primárního
- záložní: proxy server, který přeposílá dotazy dalším DNS servrům

DNS rezoluce

Rezoluce je proces hledání v odpovědi v systému DNS. Vyhledávání začíná dotazem na kořenový server.

Pokud je dotaz rekurzivní, musí dotazovaný server odpovědět požadovanými daty, nebo chybovou hláškou. Daná data buď zná, nebo se na ně zeptá dalšího serveru.

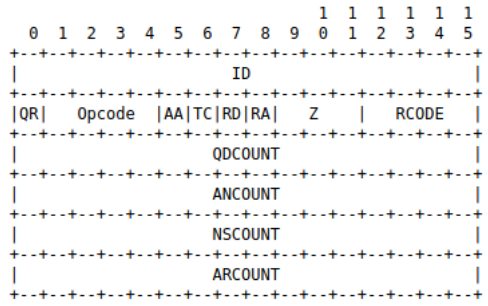
Pokud je dotaz iterativní, vrací buď požadovaná data, protože je má u sebe, nebo adresu servru, na který se máme dotázat.

Struktura DNS zprávy [2]

Hlavička

Z hlavičky nás zajímá především:

- QR: určuje, zda se jedná o dotaz, či odpověď
- RCODE: kód určující úspěšnost paketu (např. not implemented, refused, ...)
- QCOUNT: počet dotazů obsažených ve zprávě

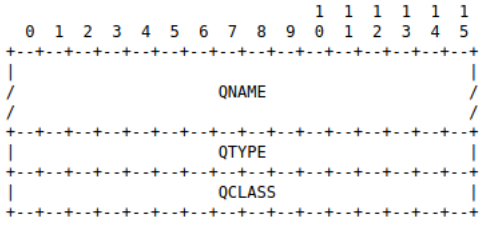


Obrázek 1: DNS hlavička

Tělo dotazu

Hlavní položkou je QNAME, která obsahuje název poddomény. Název je rozdělen na `labels`, což jsou čáti domény, které by v textové podobě byly odděleny tečkou. Každý `label` je uvozen jedením oktetem specifikující počet znaků daného `labelu`. Konec domény je označen nulovým oktetem.

QTYPE specifikuje typ dotazu (např. A, AAA, MX, ...) a QCLASS třídu dotazu(např. IN pro internet).



Obrázek 2: DNS tělo

2.2 DNS filtering

DNS filtering je metoda využívající DNS k blokování překladu adres, které mohou být pro uživatele potenciální hrozbou, vést na stránky se škodlivý obsahem, nebo narušovat soukromí.[4]

DNS filter je server, který zachytává a analyzuje DNS požadavky. Pokud zjistí, že daný požadavek chce přeložit zakázanou doménu, pošle zpět vhodnou odpověď jako IP adresu localhosta, IP adresu stránky informující o blokovaném obsahu, nebo chybu.

3 Návrh implementace

Dle zadání je třeba vytvořit UDP DNS server, který bude podporovat IPv4 i IPv6. Server bude filtrovat požadavky a ty validní přeposílat na specifikovaný DNS resolver.

4 Implementace

K implementaci byl zvolen jazyk C++, protože umožňuje objektově orientované programování a lehčí práci s poli. Ze síťových knihoven jsem využila: netinet, arpa a sys. Pro práci s vlákny pak unistd.

4.1 Moduly

Celý projekt byl rozložen do několika modulů.

main

Vstupní modul, kde se zpravovávají vstupní parametry, inicializují se třídy a spouští se server. Modul má taktéž za úkol úklid po ukončení serveru.

logger

Modul umožňující 3 základní úrovně logování (disabled, verbose a debug).

ErrorException

Tento hlavičkový soubor obsahuje definice všech použitých výjimek. Výjimky obsahují i metodu `exit_with_code()`, která vypíše chybovou hlášku a skončí s návratovou hodnotou odpovídající dané chybě.

DomainLookup

Třída, která obstarává vyhledávání filtrovaných domén v zadaném souboru.

DNSFilter

Tato třída je jádrem celého projektu. Vytváří server, který zpracovává DNS požadavky na daném portu a zprostředkovává odpověď.

4.2 Parametry

-s server [-p port] -f filter_file [-v]

- -s: IP adresa nebo doménové jméno DNS serveru (resolveru), kam se má přeposlat dotaz. Verze IP adresy serveru se shoduje s verzí IP adresy resolveru. Pokud je zadané doménové jméno, pak se z něj odvodí IPv4 adresa.

- -p: Číslo portu, na kterém bude program očekávat dotazy. Výchozí je port 53.
- -f: Jméno souboru obsahující nežádoucí domény. Soubor může obsahovat komentáře (#) a prázdné řádky.
- -v: Výpis informací

4.3 Server

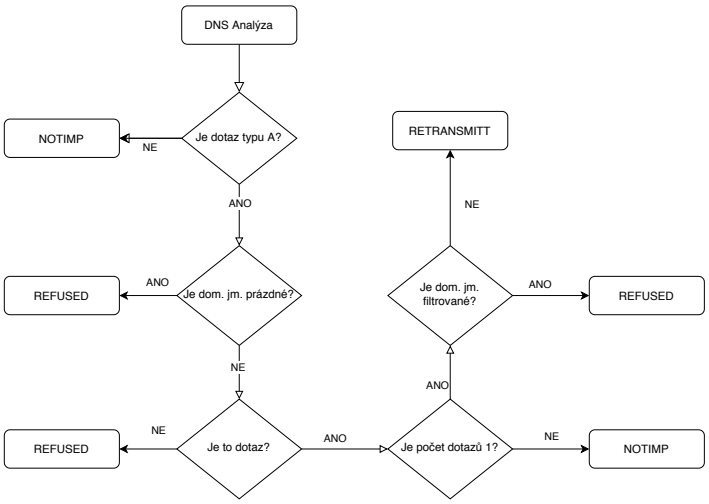
Implementovaný server je UDP server běžící na zadaném portu. Verze IP adresy, kterou server používá, je daná verzí IP adresy vnějšího DNS servru, na který jsou přeposílány validní požadavky. IP adresa je adresa localhosta.

Každý požadavek, který na server příjde, je zpracován zvlášť ve svém vlastním vlákně, proto nehrozí hromadění požadavků při vyšší zátěži. V tomto samostatném vlákně probíhá analýza DNS zprávy a získání odpovědi.

Pokud se analýzou zjistí, že DNS dotaz není z nějakého důvodu validní, tak se zpět posílá odpověď vygenerovaná přímo na servru. V opačném případě se vytvoří DNS klient, který přepošle dotaz na specifikovaný DNS server a odpověď vrátí svému klientovy. Při komunikaci s DNS servrem je nastaven timeout na 2s.

4.4 Analýza

Analýza probíhá nad DNS zprávou a na jejím základě se rozhoduje, jakým způsobem se zpracuje odpověď. Následující obrázek ukazuje postup při rozhodování.



Obrázek 3: Analýza diagram

Pro analýzu DNS hlavičky byla vytvořena struktura dns_header, která je implementována podle RFC 1035. Pořadí prvků ve struktuře je odlišné pro big indian a little indian systémy, program tedy funguje na obou typech.

4.5 Filtrování domén

Jak už bylo zmíněno, kontrola domén oproti filtrovacímu souboru je implementována ve třídě `DomainLookup`. Pro uložení filtrovaných domén byl zvolen datový typ `unordered map`. Ten obsahuje pouze unikátní hodnoty a je implementován jako hešovací pole, což znamená, že operace vložení, vyhledávání a odstranění probíhají v konstantním čase [3].

4.6 Sig term

V programu se pracuje s vlákny, inicializují třídy a pracuje se sokety a soubory. Je tady třeba korektně ukončit program v případě zavolání signálu `SIG TERM`. Tuto obsluhu signálu zajišťuje statická metoda `sigterm_handler` třídy `DNSFilter`.

Server běží ve smyčce kontrolující statickou proměnou `run` a stav návratové hodnoty blokující funkce `recvfrom`. V metodě se tedy nastaví proměnná `run` na false a zároveň se zavolá funckce `shutdown` na daný soket, který ukončí i běh `recvfrom`.

4.7 Chybové návratové kódy

kód	popis
1	špatné argumenty
2	nelze otevřít filtrovací soubor
3	problém se soketem
4	nelze přeložit IP adresu, nebo doménové jméno

4.8 Testování

Součástí projektu jsou unit testy pokrývající třídu DomainLookup a statické metody třídy DNSFilter. Dále pak y/`bash` skript, který pokrývá funkcionalitu spuštěného servru pomocí `nslookup`.

5 Použití

Pro ilustraci je použit nástroj nslookup.

Vytvoření příkladu filtrovacího souboru

```
echo "facebook.com" > filter.example;
```

Spuštění dns filtru.

```
./dns -s 1.1.1.1 -p 1234 -f filter.example
```

Odeslání validního dotazu pomocí nástroje nslooup.

```
nslookup -port=1234 -type=A  google.com localhost
Server:                 localhost
Address:        127.0.0.1#1234

Non-authoritative answer:
Name:        google.com
Address: 216.58.201.110
```

Odeslání dotazu jiného typu než A

```
nslookup -port=1234 -type=AAAA  google.com localhost
Server:                 localhost
Address:        127.0.0.1#1234

** server can't find google.com: NOTIMP
```

Odeslání dotazu na filtrovanou doménu

```
nslookup -port=1234 -type=A  add.facebook.com localhost
Server:                 localhost
Address:        127.0.0.1#1234

** server can't find add.facebook.com: REFUSED
```

Zdroje

[1] [online]. Dostupné na: https://cs.wikipedia.org/wiki/Domain_Name_System.

[2] [online]. Dostupné na: https://www.ietf.org/rfc/rfc1035.txt.

[3] [online]. Dostupné na: https://en.cppreference.com/w/cpp/container/unordered_set.

[4] *DNS Filtering* [online]. Dostupné na: https://www.titanhq.com/a-guide-to-dns-filtering-under-

[5] MATOUŠEK, P. *Síťové aplikace a jejich architektura*. [b.m.]: VUTIUM, 2014.