Vysoké učení technické v Brně

Fakulta informačních technologií



Dokumentace k projektu do předmětu ISA

DNS server

20. listopadu 2016

Autor: Filip Března, xbrezn00@stud.fit.vutbr.cz

Fakulta informačních technologií

Vysoké učení technické v Brně

Obsah

1.	Důl	ežité pojmy	1
	1.1.	Domain Name System	1
	1.2.	DNS server	1
	1.3.	Komunikační zprávy pro DNS server	2
		rh programu	
		lementace	
	3.1.	Použité knihovny	3
4.	Moż	žnosti aplikace	4
5.	Záv	ěr	4
	Metriky kódu		
	Reference		

Úvod

Dokumentace do předmětu Síťové aplikace a správa sítí na téma DNS server se zabývá, jak již název napovídá DNS serverem. Vysvětluje pojem DNS a DNS server a přibližuje danou problematiku.

1. Důležité pojmy

Tato kapitola popisuje návrh a implementaci programu DNS server. Bude zde přiblížen princip a význam tohoto tématu a také implementace a výsledek práce.

1.1. Domain Name System

DNS (Domain Name System), tedy systém doménových jmen. Jeho hlavní funkcí je vzájemný převod doménových jmen a IP adres jim náležícím. Důvod převodu IP adres na doménová jména je prostý, má usnadnit uživatelům práci s adresami s kterými se v textové podobě lépe manipuluje. Avšak mezi sekundární činnost můžeme zařadit i funkce pro elektronickou poštu a IP telefonii.

Protokol DNS běží na portech 53 a umožňuje režii nad IPv4 i IPv6 adresami. Prostor doménových jmen je tvořen stromem s jedním kořenem a každý uzel tohoto stromu obsahuje informace o doméně a odkazy na podřízené domény

1.2. DNS server

DNS server někdy označovaný jako *nameserver*, je server poskytující službu DNS. DNS server naslouchá požadavkům klientů a odpovídá na ně pomocí předem daných pravidel. Server vyhledává data pro odpovědi v tzv. zónových souborech a v případě, že nebyla nalezena dojde k rekurzivnímu dotazu na vzdálený DNS server.

1.3. Komunikační zprávy pro DNS server

```
▼Domain Name System (query)
  [Response In: 448139]
Transaction ID: 0xc70e
▼Flags: 0x0100 Standard query
   0... .... = Response: Message is a query
   .000 0... .... = Opcode: Standard query (0)
   .... ..0. .... = Truncated: Message is not truncated
   .... ...1 .... = Recursion desired: Do query recursively
   .... = Z: reserved (0)
   .... .... ...0 .... = Non-authenticated data: Unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
▼Queries
 ▼google.com: type A, class IN
    Name: google.com
    Type: A (Host address)
    Class: IN (0x0001)
```

Obrázek 1: Ukázka požadavku query ze síťového analyzátoru Wireshark

```
▼Domain Name System (response)

  [Request In: 448138]
  [Time: 0.041805000 seconds]
 Transaction ID: 0xc70e
▼Flags: 0x8180 Standard query response, No error
   1... = Response: Message is a response
   .000 0... .... = Opcode: Standard query (0)
   .... .0.. .... = Authoritative: Server is not an authority for domain
   .... ..0. .... = Truncated: Message is not truncated
   .... ...1 .... = Recursion desired: Do query recursively
   .... 1... = Recursion available: Server can do recursive queries
   .... = Z: reserved (0)
   .... ..... .0. .... = Answer authenticated: Answer/authority portion was not authenticat
   .... .... Unacceptable
   .... .... 0000 = Reply code: No error (0)
  Questions: 1
 Answer RRs: 1
 Authority RRs: 0
 Additional RRs: 0
▼ Queries
 ▼google.com: type A, class IN
    Name: google.com
    Type: A (Host address)
    Class: IN (0x0001)
▼ Answers
 ▼google.com: type A, class IN, addr 216.58.201.110
    Name: google.com
    Type: A (Host address)
    Class: IN (0x0001)
    Time to live: 4 minutes, 59 seconds
    Data length: 4
    Addr: 216.58.201.110 (216.58.201.110)
```

Obrázek 2: Ukázka odpovědi response ze síťového analyzátoru Wireshark

2. Návrh programu

Jedná se o síťovou aplikaci a ta tedy pracuje s BSD sockety. Posílány jsou pouze IPv4 packety a zpráva pracuje na transportní vrstvě s UDP protokolem, jelikož není nutné spolehlivé doručování zpráv. Server není navržen pro současnou obsluhu více klientů, avšak komunikace je dostatečně rychlá, a proto není nezbytně nutné implementovat tuto službu. Server odpovídá pouze na požadavky typu A a MX, které reprezentují dotaz na IPv4 adresu, což je velice klíčové, resp. dotaz o poštovních serverech. Server hledá odpovědi v zónových souborech. Jelikož nebyla nalezena solidní knihovna pro parsování zónových souborů, využívá se vlastních funkcí.

3. Implementace

Program byl implementován v programovacím jazyce C++ a otestován na překladačem g++ 4.8.4 a je nutné jej spouštět s právy uživatele root.

3.1. Použité knihovny

```
#include <iostream>
#include <stdio.h>
#include <cstdlib>
#include <string>
#include <stdint.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <iomanip>
```

4. Možnosti aplikace

-m [--mitm] <ip address> resolvuje každý dotaz na A na zvolenou IP

-h [--help] vypíše nápovědu

-p [--port] port na kterém bude server naslouchat, pokud parametr není zadaný,

použije standardní port 53

<zonefile> posledním povinným argumentem je jméno zónového souboru

Syntax: ./roughDNS [-m] [-h] [-p <port>] [<zonefile>]

5. Závěr

Program přijímá žádosti typu A a MX a posílá na ně patřičné odpovědi ze svého zónového

souboru v případě nálezu, ze zónového souboru vzdáleného DNS serveru v případě, že nemohlo

být žádosti vyhověno. DNS server obsluhuje v daný okamžik pouze jednoho klienta. Program

je překládán pomocí překladače g++ a slouží k tomu soubor Makefile.

Metriky kódu

Počet souborů: 8

Počet řádků zdrojového textu pro DNS server: 704

Počet řádků zdrojového textu pro parser zónových souborů: 753

Velikost spustitelného souboru: 121,6 kB

Reference

[1] RFC 1035 [online]. 1987, stran 42 [cit. 2016-11-20].

Dostupné z: https://www.ietf.org/rfc/rfc1035.txt

4