VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií
Faculty of Information Technology

ISA

DNS RESOLVER

Brno, 2019 Lukáš Havlíček (xhavli46)

Obsah

1 Popis programu

Program slouží pro zasílání dotazů na DNS server a zpracování následné odpovědi. Program byl napsán v jazyce C++ pomocí BSD socketů. Program zasílá na základě argumentů dotazy na server pomocí protokolu UDP.
Hlavní účel DNS serveru je překlad mezi doménovými jmény a ip adresami. Kde pro získání ipv4 adresy se využívá záznamu A, pro ipv6 adresu záznam AAAA a pro překlad ip adresy na doménové jméno záznamu PTR.

2 Implementace

Jak jsem již psal v popisu programu, program využívá po zasílání dotazů BSD sockety, přesněji funkce sendto() a recvfrom(). Díky využití protokolu UDP (sock_dgram) není potřeba vytvářet ip header ani UDP header, o to se stará knihovna. V programu je tedy potřeba pouze vytvořit DNS header + data. Stejně tak při zpracování odpovědi se jedná o zpracování od DNS headeru.

2.1 Argumenty

Pro zpracování argumentů jsem se rozhodl využít cyklu přes argumenty programu, kde každý argument je porovnán se známým argumentem. Díky tomu nezáleží na pořadí argumentů, ale na druhou stranu se nedají parametry „spojovat“, tím pádem nelze například spustit program s přepínačem „-6r“, ale je potřeba uvést přepínače odděleně tj. „-6 –r“.

Dle zadání lze program spustit následujícím stylem:

./ dns [-r] [-x] [-6] -s server [-p port] adresa

Kde:

-r znamená vyžadování rekurze, tím pádem je v DNS headeru nastaven RD(recursion desired) flag na 1. Odpověď ale nemusí být rekurzivní, pokud server rekurzi nepodporuje.

-x znamená reverzní dotaz namísto přímého, na rozdíl od klasického dotazu jde zde adresa zkontrolována, jestli se jedná o platnou ipv4 adresu (pokud je –x zadán současně s -6 přepínačem tak ipv6 adresu). Tato adresa je následně „převrácena“ a s příponou .in-addr.arpa (.ip6.arpa) zaslána jako PTR dotaz.

-6 specifikuje zda se má jednat o AAAA(ipv6) záznam namísto výchozího A(ipv4). V případě kombinací s –x určuje verzi zadané adresy.

-s server udává server, na který se má dotaz zaslat. Server může být zadán jako ipv4/ipv6 adresa nebo jako doménové jméno (v případě doménového jména je využita funkce getaddrinfo() pro překlad). Na rozdíl od –x parametru program sám rozpozná, zda se jedná o ipv4 nebo ipv6 adresu (nebo doménové jméno).

-p port udává port, na který se má dotaz zaslat. Výchozí hodnota je 53. Je možno zadat port z rozsahu 1 až 65535.

adresa udává obsah dotazu zaslaného na server. Pokud se jedná o reverzní dotaz je převedena na PTR dotaz (viz popis parametru –x). Jinak je převedena do notace kde jsou tečky nahrazeny délkou textu co následuje za tečkou. Například www.fit.cz by byl nahrazen za „\3www\3fit\2cz“. Maximální délka adresy je 253 znaků.

2.2 Zaslání a přijímání odpovědi

Program využívá pro komunikaci se serverem funkce sendto() a recvfrom(), kterým je předána struktura obsahující informace o server na který se má dotaz poslat. V případě neúspěchu je program ukončen a je vypsána chybová hláška z errno.
Vzhledem k tomu že je využíváno UDP protokolu, který je „nespolehlivý“, tak nemusí od serveru přijít odpověď a funkce recvfrom() by čekala „do nekončena“. Proto je před touto funkcí nastaven alarm na 3s a pokud nestihne přijít odpověď do 3s, tak je program ukončen s chybovou hláškou říkající, že vypršela čekací doba.

2.3 Reverzní dotaz

Jak bylo popsáno u popisu argumentů, tak je využíváno PTR dotazů. Kde je zadaná adresa zkontrolována, jestli se jedná o validní ipv4 adresu (s parametrem -6 ipv6 adresu). Následně je adresa upravena do formátu pro poslání.
Například adresa 8.8.4.4 bude převedena do formátu:
4.4.8.8.in-addr.arpa
nebo adresa 2001:db8::567:89ab by byla převedena do formátu:
b.a.9.8.7.6.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa (viz [1] a [2])
Jinačí možnost reverzního dotazu je nastavení OPCODE v DNS headeru na 1 tj. inverse query. A záslání dotazu v answer části, a přečtení odpovědi v question části. Ale tento způsob byl „zrušen“ v roce 2002 viz. [3]. A je nyní využíváno PTR záznamů.

2.4 Podporované formáty

Program plně podporuje a vypisuje všechny údaje formátů: A, AAAA, NS, CNAME, SOA, PTR. Pokud náhodou přijde neznámy formát, jsou data vypsána v hexadecimálním formátu + je vypsán název typu (pokud byl definován v RFC 1035 + AAAA), jinak je vypsána pouze jeho číselná hodnota. U class je podporován typ pouze IN (pokud přijde jiný je vypsán stejným stylem jako type).

3 Testování

Testování probíhalo pouze manuální. Kde jsem využíval převážně programu wireshark, pro kontrolu odeslaných a přijatých zpráv. Testování probíhalo ve virtuálním prostředí, přesněji na image, které nám bylo poskytnuto minulý rok pro projekt do předmětu IPK jako referenční image pro síťové předměty. Následně jsem hotový program otestoval na serverech eva a merlin.

Příklad spuštění:

./dns -r -s kazi.fit.vutbr.cz www.fit.vut.cz

Příklad výstupu:

Sending dns query to: 147.229.8.12

Authoritative: Yes, Recursive: Yes, Truncated: No, Reply code: 0(OK)

Question section(1)

www.fit.vut.cz, A, IN

Answer section(1)

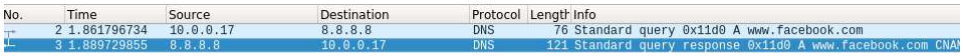
www.fit.vut.cz, A, IN, 14400, 147.229.9.26

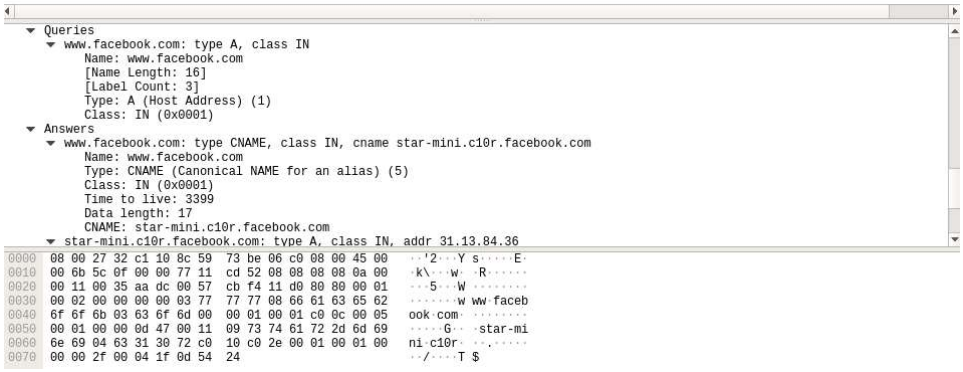
Authority section(0)

Additional section(0)

Program navíc vypisuje ip adresu serveru kam zasílá dotaz a Reply code obsažen v DNS headeru, převážně pro případ kdy se liší od 0(OK).

Ukázka z wiresharku pro ./dns -s 8.8.8.8 www.facebook.com:





Literatura

[1] RFC 1035[online]. Dostupné na URL: < https://tools.ietf.org/html/rfc1035>

[2] RFC 3596[online]. Dostupné na URL: < https://tools.ietf.org/html/rfc3596>

[3] RFC 3425[online]. Dostupné na URL: < https://tools.ietf.org/html/rfc3425>