Vysoké učení technické v Brně, Fakulta informačních technologií

ISA – Projekt

Varianta: Export DNS informací pomocí protokolu Syslog

Vypracoval Pavel Janko (xjanko10)

Obsah

Uvedení do problematiky

Každý počítač má nastaven jmenný server, který používá pro překlad jmen. Když uživatel počítače použije doménové jméno tím, že jej napíše například do řádku adresy ve webovém prohlížeči, počítač pošle dotaz, jaká IP adresa odpovídá tomuto doménovému jménu, jmennému serveru. Ten buď dotaz zodpoví přímo, pokud dotazové doménové jméno zná, anebo dotaz předá dál, dalším jmenným serverům v doménovém stromu.[1]

Ke snímání těchto dotazů a odpovědí se využívají analyzátory paketů. V okamžiku, kdy datové proudy tečou přes síť, tak analyzátor paketů zachycuje každý paket. Ten je poté v případě potřeby dekódován na surová data paketu, která ukazují hodnoty různých polí v paketu. Dále pak je analyzován jeho obsah podle příslušné RFC nebo jiných specifikací.[2]

Základní informace o programu

Cílem tohoto projektu je vytvořit právě takový analyzátor paketů, který bude zpracovávat pakety protokolu DNS. Určité typy těchto zpráv, konkrétně pouze odpovědi, následně rozebere, převede z datového formátu do formátu specifikovaného RFC pro zprávy Syslog, přičemž zpráva bude obsahovat informace v smysluplném formátu, a následně podle kombinace argumentů buď vypíše na výstup, nebo odešle na specifikovaný Syslog server.

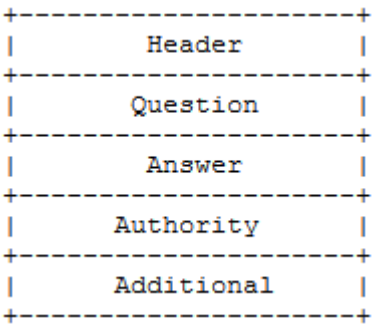
1 https://www.nic.cz/page/312/o-domenach-a-dns/

2 https://cs.wikipedia.org/wiki/Analyz%C3%A1tor_paket%C5%AF

Přehled nastudovaných informací

Zpráva

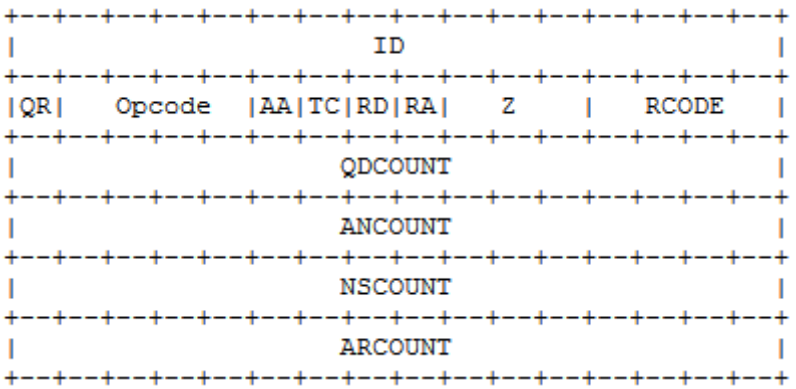
Zpráva slouží jako základní komunikační jednotka doménového protokolu a má dle RFC



1035[3] následující strukturu:

1. Hlavička – je vždy přítomná a obsahuje položky které specifikují, které ze zbývajících sekcí se ve zprávě vyskytují; rovněž obsahuje informaci o tom, zdali je zpráva dotaz nebo odpověď apod., více níže
2. Otázka – specifikuje typ dotazu (*QTYPE)* a název domény dotazu (*QNAME*)
3. Odpověď – obsahuje zdrojové záznamy, které odpovídají na otázku
4. Autorita – obsahuje zdrojové záznamy, které ukazují směrem k autoritativním jmenným serverům
5. Dodatečné – obsahuje zdrojové záznamy, které se vztahují k dotazu, ale nejsou přímo odpovědí na tento dotaz

Hlavička



Obsahuje následující pole:

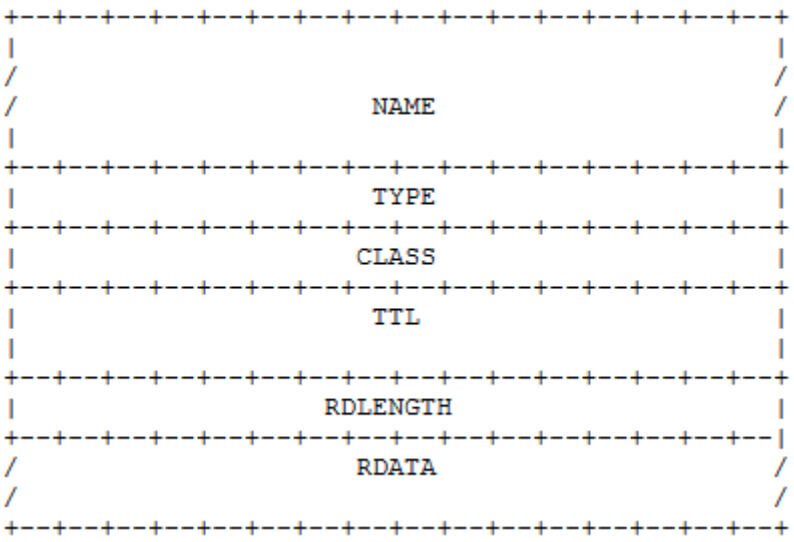
1. ID – šestnáctibitový identifikátor přiřazený programem, který generuje dotaz
2. QR – jednobitové pole specifikující, zdali je zpráva dotaz nebo odpověď
3. OPCODE – čtyřbitové pole, které specifikují druh dotazu (v programu se používá pouze typ „0“, tedy standardní)

---

3 https://www.ietf.org/rfc/rfc1035.txt

4. AA – jedná se o bit, který je validní tehdy, když na dotaz odpovídá jmenný server, který je autoritou pro doménové jméno v dotazu
5. TC – tento bit je validní, pokud byla zpráva zkrácena například kvůli limitům UDP na délku 512
6. RD – tento bit určuje, zdali je žádoucí dotaz zpracovávat rekurzivně
7. RCODE – čtyřbitové pole, které specifikují stav zprávy po přijetí (nechybovost, případně druh chyby)
8. QDCOUNT – šestnáctibitové pole reprezentující počet dotazů v sekci dotazů
9. ANCOUNT – šestnáctibitové pole reprezentující počet zdrojových záznamů v sekci odpovědí
10. NSCOUNT – šestnáctibitové pole reprezentující počet záznamů jmenných serverů v sekci autoritativních záznamů
11. ARCOUNT – šestnáctibitové pole reprezentující počet zdrojových záznamů v sekci dodatečných záznamů

Zdrojový záznam



Má následující strukturu:

1. NAME – doménové jméno, které se ke zdrojovému záznamu vztahuje
2. TYPE – šestnáctibitové pole obsahující jeden z kódu pro typ záznamu
3. CLASS – šestnáctibitové pole které specifikují třídu dat v datovém poli
4. TTL – 32 bitové pole, které specifikuje časový interval, po který je možné zdrojový záznam mít uložen ve vyrovnávací paměti před tím, než jej bude potřeba zneplatnit
5. RDLENGTH – šestnáctibitové pole specifikující počet oktetů v datovém poli
6. RDATA – řetězec různého počtu oktetů, který popisuje zdroj – formát se liší podle type a třídy specifikované výše

Komprese zpráv

Za účelem snížení velikosti zpráv využívá doménový systém kompresní schéma, které vylučuje duplikaci doménových jmen ve zprávě. Schéma umožňuje, aby se doménové jméno ve zprávě skládalo buď ze sekvence návěstí končících nulovým oktetem, ukazatelem anebo sekvencí návěstí končících ukazatelem.

Ukazatele lze využít pouze pro výskyt doménového jména, jejichž formát není specifický pro třídu. Konkrétní řešení komprese ať už odesílaných zpráv nebo dekomprese těch přijímaných lze vyčíst buď ze zdrojového kódu, nebo případně v kapitole „Popis zajímavých pasáží implementace“.

Návrh aplikace

Jednotlivé části programu jsou rozděleny do souborů podle toho, do které části životního cyklu DNS dotazu (a následné odpovědi) jsou zapojeny.

Header.cpp/h

Tato část programu se stará o zpracování hlavičky odpovědi na DNS dotaz, kdy se například ověřuje, že je platný kód odpovědi apod.

Hlavička je implementována jako struktura, a to především proto, že jednotlivé její části musí být přesně řazeny za sebou co se týče bitové posloupnosti dat, spolu s odpovídajícími velikostmi – tato vlastnost lze obtížněji zaručit při využití tříd.

Question.cpp/h

V této struktuře se řeší určení typu dotazu a také třída dotazu – používá se „1“ pro internet.

Rovněž se v ní ověřuje (skrze konstruktor), zdali argument typu dotazu specifikovaný uživatelem je zpracovatelný programem.

Record.cpp/h

Nejdůležitější část programu, která zpracovává záznamovou část dotazu, tedy tu, kterou k dotazu připojil DNS server při odpovědi.

Mezi její hlavní funkce patří převod IP adres ze síťové formy do řetězcové a rovněž převod z prefixové formy názvů hostitelů (návěstí apod.) na řetězcovou.

Pro každou odpověď, autoritativní jmenný server nebo dodatečný záznam je vytvořena nová instance této třídy.

Query.cpp/h

Jedná se o třídu, která se stará o obdržení odpovědi spolu se záznamy z DNS serveru.

Rovněž v sobě ve formě atributů obsahuje hlavičku a otázku, které používá právě pro dekonstrukci DNS odpovědi, také obsahuje vektor odpovědí, autoritativních jmenných serverů a dodatečných záznamů.

main.cpp

Kořenová funkcionalita programu, která zajišťuje zpracování argumentů z příkazové řádky a logiku pro zpracování ať už pcap souboru anebo real-timeové odchytávání komunikace na

rozhraní ve smyčce. Také obstarává výpis odpovědí na výstup, případně odeslání odpovědí na zadaný Syslog server.

Zároveň se stará o výpis chybového hlášení na standardní chybový výstup a také o ukončení programu s korektním kódem.

Popis zajímavých pasáží implementace

Převod z prefixového tvaru (ukazatel) na řetězec

Jelikož může ukazatel odkazovat na jiný ukazatel, tak jsem se rozhodl funkcionalitu pro sestavení řetězce z ukazatelů (ale i bez nich) implementovat rekurzivně (konkrétně ve funkci *void ParseName* v souboru *Record.cpp*). Nejprve se podle prvních dvou bitů prvního oktetu části jména zjistí, zdali se vůbec jedná ukazatel nebo ne (musí být nulové). Komplikace však nastává v případě, kdy se ukazatel objevuje až na konci jména – program tuto skutečnost musí být schopen detekovat.

Obecně tedy funkce funguje tak, že na konec předpřipraveného řetězce určeného pro přeložení jména ze síťové formy přidává všechny znaky, na které narazí – znaky pozná tak, že se v prvním oktetu nevyskytují na prvních dvou bitech nuly. V opačném případě, kdy se jedná o ukazatel, tak funkce volá sama sebe s tím, že se index v bufferu mění na pozici, na kterou ukazuje právě nalezený ukazatel. Funkce s převodem končí v ten moment, kdy narazí na nulový oktet.

Odeslání na server/výpis na standardní výstup po uplynutí času

Vzhledem k tomu, že k vypršení doby určené pro vyexportování statistik může dojít i v době, kdy se zpracovávají pakety, bylo nutné zajistit neblokující komunikaci při odposlouchávání zavoláním funkce *pcap_setnonblock()*[4].

Následně se oproti typickému využití knihovny *pcap.h* nevyužila funkce *pcap_loop()*[5], díky které by se rovnou skočilo do callback metody a z ní by se nikdy neodešlo, nýbrž funkce *pcap_dispatch()*[6], která zpracuje pouze jeden paket. Ta je volána v cyklu, jehož podmínka platí pouze do doby, než uplyne časovač, poté se provede výpis a časovač se zresetuje.

4 http://man7.org/linux/man-pages/man3/pcap_setnonblock.3pcap.html
5 https://linux.die.net/man/3/pcap_loop
6 https://linux.die.net/man/3/pcap_dispatch

Návod pro použití programu

Program lze spustit po přeložení příkazem *make*. Možná kombinace parametrů je specifikována níže.

```
Použití: ./dns-export [-h]
         ./dns-export [-r file.pcap] [-i interface] [-s syslog-server] [-t seconds]
Popis parametrů:
         -h (help) - volitelny parametr, pri jeho zadani se vypise napoveda a
program se ukonci
         -r (file.pcap) - volitelny parametr, pcap soubor, ktery se ma zpracovat
         -i (interface) - volitelny parametr, na kterem rozhrani se ma naslouchat na
DNS provoz
         -s (syslog-server) - volitelny parametr, hostname (nebo IPv4/v6) adresa
syslog serveru
         -t (seconds) - volitelny parametr, doba vypoctu statistik v sekundach,
(vychozi hodnota 60 sekund)
```

Program rozlišuje mezi dvěma druhy chyb:

1. *runtime_error*[7]
   - tato chyba vzniká při zpracování neočekávaných událostí, jako je přijetí nekorektního návratového kódu odpovědi na dotaz, nebo například dosažení limitu rekurze při zpracování návěstí
   - na vznik těchto chyb nemá uživatel vliv a vrací návratový kód „1“
2. *invalid_argument*[8]
   - tato chyba vzniká důsledkem neplatného vstupu zadaného uživatelem, nebo v případě, že je zadána neplatná kombinace parametrů
   - v případě výskytu tohoto druhu chyby vrací program návratový kód „2“

Použití programu

```
root@laptop:~/Projekty/isa/cmake-build-debug# ./dns-export -i any -t 10
<134>1 2018-11-18T20:43:30.747Z 192.168.10.170 dns-export - - - google.com A 216.58.201.110 2
<134>1 2018-11-18T20:43:30.749Z 192.168.10.170 dns-export - - - google.com AAAA 2a00:1450:4014:801::200e 2
<134>1 2018-11-18T20:43:33.429Z 192.168.10.170 dns-export - - - seznam.cz A 77.75.79.53 2
<134>1 2018-11-18T20:43:33.436Z 192.168.10.170 dns-export - - - seznam.cz AAAA 2a02:598:a::79:39 2
<134>1 2018-11-18T20:43:30.747Z 192.168.10.170 dns-export - - - google.com A 216.58.201.110 2
<134>1 2018-11-18T20:43:30.749Z 192.168.10.170 dns-export - - - google.com AAAA 2a00:1450:4014:801::200e 2
```

7 http://en.cppreference.com/w/cpp/error/runtime_error
8 http://en.cppreference.com/w/cpp/error/invalid_argument