Vysoké učení technické v Brně
Fakulta informačních technologií

Síťové aplikace a správa sítí

Export DNS informací pomocí protokolu Syslog

Miroslav Karpíšek (xkarpi05)

17. listopadu 2018

Obsah

1 Úvod

Tento projekt je vypracován na základě zadání v předmětu *ISA - Síťové aplikace a správa sítí*. Zadání stanovuje vytvořit program pro příkazovou řádku, který umí zpracovávat příchozí šíťové pakety obsahující DNS odpovědi, které mají být analyzovány a následně formátovány do specifikovaného výstupního formátu. Hlavním cílem projektu je získat zkušenosti se získáváním smysluplných dat ze šíťových protokolů a získání povědomí o existenci standardu Syslog pro správu sítí.

2 Problematika

Aplikační protokol DNS slouží k překladu (nejen) doménových adres na IP adresy. Systém doménových jmen je hierarchicky tvořen DNS servery a lze s ním (a v rámci něj) komunikovat za použití stejnojmeného protokolu na aplikační vrstvě. Komunikace probíhá majoritně pomocí transportní vrstvy UDP, přestože standardy umožnují i komunikaci přes TCP, ta se využívá zejména v případech kdy odpovědi přesahují maximální velikosti datového rámce UDP.

Systém DNS umožnuje získávat různé informace z téže doménové adresy pomocí specifikace typu v dotazu. Díky tomu je možné různé služby mapovat na různá koncová zařízení zaštítěné stejným doménovým jménem.

Syslog je používán pro zasílání a analýzu programových dat. Často pro účely monitorování, nebo zabezpečení sítě. Formát zprávy je popsán ve standartu RFC 5424 [2].

3 Návrh

Aplikaci lze rozdělit logickým návrhem do vícero vzájemně komunikujících modulů. Jejich interakce je při použití definována uživatelem pomocí programových přepínačů. Jednotlivé moduly obstarávají tyto úkony:

3.1 Odposlech a analýzu

Řídící funkcionalita programu, která se stará o zachytávání, případně načítání jednotlivých paketů a následně provádí jejich analýzu čímž získává informace ohledně komunikace pomocí DNS protokolu. Tato analýza vychází ze znalostí komunikačních protokolů popsaných ve vícero standartech RFC [3] [1].

3.2 Formátování a ukládání statistik

Získaná data je třeba řádně formátovat tak aby jejich výstup byl smysluplný a čitelný pro člověka. Takto formátováná data je třeba ukládat do statistik, které budou později programem vytisknuty případně odeslány na syslog server. Přesný formát zpracovaných dat je definován zadáním projektu.

3.3 Komunikace se syslog serverem

Program musí být schopen komunikovat se vzdáleným syslog serverem. Tudíž jeho povinností je možnost odesílání paketů po síťi. Dále pak je třeba formátované statistiky určené k odeslání opatřit syslog hlavičkou která odpovídá výše zmíněnému standartu RFC 5424.

3.4 Obsluha signálů

Uživatel má možnost s programem komunikovat pomocí signálů. Tudíž aplikace musí být shopna signály zachytávat a vyhodnocovat tak aby buď odeslala data na server, nebo je vytisknula na standartní výstup.

4 Popis implementace

Aplikace je implementována v jazyce C++, aplikace primárně nevychází z objektově orientovaného paradigmatu, nicméně používá některé třídy a metody zahrnutých knihoven. Aplikace je členená do logických celků jež spolu vzájemně komunikují pomocí procedur.

Aplikace nejprve vyhodnotí vstupní přepínače uživatele a nastaví jednotlivé globální značky které lze číst kdekoliv v programu. Následně buď začne zachytávat pakety na zvoleném rozhraní, nebo jednotlivé pakety načítá ze zadaného vstupního souboru formátu `pcap`. Pomocí funkcí z knihovny `libpcap` nejprve ustanoví odposlech a nastaví jej pouze na pakety, které přišly na port 53, jelikož nás zajímají pakety nesoucí informace pomocí protokolu DNS.

Následně vždy při obdržení nového paketu volá *callback* funkci pro jeho zpracování. Paketem rozumnějme posloupnost oktetů obsahují informace v binární podobě. Tato funkce je řídícím prvkem celého zpracování a postupně prochází jednotlivé oktety a mapuje jejich význam na jednotlivé struktury reprezentující protokoly na vrstvách L1 až L5. V případě, že obdží nepotřebné informace tak daný paket zahodí a ukončí se. V opačném případě informace ukládá do globálního vektoru struktur které jsou později v programu využity k formátování výstupu.

Odděleným modulem je ošetření signálů, kdy se pomocí signálů indikuje jak interakce uživatele pomocí signálu `SIGUSR1` pro vypsání statistik, tak vypršení zadaného časovače pro odeslání zpráv. Modul při obdržení signálů vytvoří nový proces, který se následně postará buď o odeslání dat, nebo o jejich výpis na standartní výstup.

dalším modulem je modul starající se o správu globálních promněných, který řeší jejich kompletní správu. Globální proměnné jsem použil v případech, kdy buď z důvodu definice funkcí, které neumožnovali předat si vlastní promněnou. Nebo v případě programových parametrů, které by jinak musely být předávané skrze mnoho funkcí.

4.1 Zajímavé pasáže

- Jednotlivé hlavičky protokolů které bylo třeba v tomto projektu analyzovat jsou v mé implementaci reprezentovány strukturami. Jedním z důvodů bylo, že zejména v protokolech L1 až L3 vrstvy lze často využít vlastnosti přetypování v případě, že striktně dodržuje velikosti prvků struktury s velikostmi jednotlivých prvků hlaviček rámců jak jsou specifikovány v příslušných RFC standardech.

- Při implementaci chování na obdržení signálu jsem využil vlastností víceprocesového zpracování abych mohl během odesílání, nebo tisku statistik dále získávat a analyzovat příchozí pakety.

- Práce s TCP fragmentací obstarávám pomocí globálního vektoru struktur obsahujících nedefragmentované pakety v rámci níž probíhá celé znovusestavení TCP packetu a až při obdžení celého je ukazaztel na začátek posloupnosti oktetů předán jako ukazatel na paket. Zbytek procedury nemá tedy vůbec tušení o tom, jestli právě zpracovává paket TCP nebo UDP.

4.2 Známé nedostatky

- Kdyby uživatel poslal požadavek na vypsání dat na standartní výstup dříve, než se dokončí jejich minulé vypsání může se výstup chovat nedefinovaně, protože oba procesy by zároveň zapisovaly na stejný výstup.

- To stejné platí pro případ odeslání statistik na server, kdy ze zadání není možné odesílat statistiky v itervalech menších než 1 sekunda, což by znamenalo, že by statistiky musely být extrémě velké aby jejich odesílání trvalo více než 1 sekundu. V projektu jsem se tímto problémem nezabýval, protože by takto rychle odesílané statistky při tak velkém počtu nejpíše natolik zahltily zadaný syslog server, že by jejich využitelnost pro monitorování byla nulová.

- Sestavování větších TCP fragmentů se při *live provozu* někdy nepovede. Z časových důvodů už bohužel tento nedostatek nemohu opravit. V případě, že se tak stane, je program ukončen s chybovou hláškou. Dosavadní nasbírané statistiky se ovšem pošlou, pokud je to možné. (program se neukončí násilným ukončením `Segmentation fault`). Při testování mi sestavování TCP paketů z .pcap souboru fungovalo, při *live provozu* ovšem stejná funkcionalita obsahuje chybu. Nejspíš to bude mít co dočinění s rychlostí zpracování jednotlivých paketů.

5 Informace o programu

Program byl vyvýjen na operačním systému `macOS v10.14.1` kde byl i testován, stejně tak byl testován na referenčním stroji disponující operačním systémem `CentOS v7.5.1804`.

5.1 Typy DNS

Program dle zadání podporuje zpracování všech typů DNS záznamů které byly definovány na fóru. Jednotlivé statistiky poté vyhovují formátu který byl zadán. Kdy při nejasnostech ve výpisu určitého záznamu jsem jsem se řídil instrukcemi v adekvátních standartech RFC. Proto například u DNSKEY využívám modulu `base64` jež implementoval *René Nyffenegger*[4]. Tento převzatý zdrojový text je v mých souborech řádně okomentován a to včetně licence, která povoluje redistribuci. Učinil jsem tak z důvodu, že jsem předpokládal, že převod do formátu `base64` není hlavním cílem projektu.

5.2 Slučování zpráv pro syslog

Při implementaci jsem si nebyl jist jakým způsobem mají být odděleny jednotlivé syslog zprávy, proto jsem se uchýlil k variantě, kdy jsem umožnil pomocí přepínače tuto vlastnost programu ovládat. V případě, že je slučování zpráv zapnuta, jednotlivé syslog zprávy jsou odděleny mezerou. V základním chování programu je slučování vypnuta.

5.3 Podpora TCP fragmentace

Stejně jako v předešle podkapitole i zde jsem se raději uchýlil k možnosti tuto vlastnost programu umožnit ovládat uživatelem pomocí programového přepínače. Znovu základní chování programu TCP fragmentaci neumožňuje a je třeba je explicitně povolit.

6 Použití programu

Program funguje v různých režimech ovládáných pomocí přepínačů. Všechny z nich jsou volitelné, některé však jsou vzájemně nekompatabilní. Jejich funkčnost je také popsána v přiložené manuálové stránce, stejně tak i při spuštění programu s parametrem `-h`

`-i interface:`
: Pokud je zadán, program naslouchá na zadaném rozhraní. Nesmí být definováno společně s přepínačem -r. Na referenčním stroji podporuje i odposlech na všech rozhraních zároveň.

`-r resource:`
: V případě, že je zadaný soubor, program jej bere jako vstup příjmaných paketů. Takto zpracovávaný program vypíše/pošle statistiky až při jeho ukončení. Tudíž přepínáč -t nesmí být v kombinaci s tímto použit.

`-s syslog:`
: Specifikuje pomocí `hostname`, `IPv4` nebo `IPv6` server na který mají být odeslány statistiky ve formátu specifikovaným v RFC 5424 [2]. Statistiky jsou ve výchozím nastavení odesílány v intervalu 60 sekund. Uživatel tuto hodnotu může upravit pomocí přepínače -t

`-t timeout:`
: Hodnota tohoto přepínače udává délku prodlevy po které se odesílají data na nastavený syslog server. Hodnota musí být kladné celé číslo větší než 0

`-f`:
: Tento parametr nebyl specifikován v zadání, v případě, že je přítomen program podporuje TCP fragmentaci.

`-c`:
: Tento parametr nebyl specifikován v zadání, v případě, že je přítomen program podporuje seskupování vícero statistik do jedné syslog zprávy.

6.1 Návratové hodnoty

V případě úspěšného ukončení programu je návratovou hodnotou dle konvencí 0. V případě výskytu chyby se návratová hodnota řídí podle popisu v knihovně `<sysexits.h>`. Kdy při chybovém ukončení se uživateli vypíše chybová hláška. V případech kdy je to možné, se před samotným ukončením při výskytu chyby pokusí program odeslat/vypsat `std::out` dosavadní statistiky. Kromě posledního paketu při jehož zpracování došlo k chybě.

Dle zadání jsem pochopil, že `SIGINT` je v tomto projektu zcela legitimním vstupem uživatele (jediný validní způsob ukončení programu při odposlechu na síťovém rozhraní) proto návratová hodnota po obdržení tohoto signálu je neobvykle 0.

Reference

[1] Arends, R.: *DNS Security Introduction and Requirements*. 2005, [Online].
URL `https://tools.ietf.org/html/rfc4033`

[2] Gerhards, R.: *The Syslog Protocol*. 2009, [Online].
URL `https://tools.ietf.org/html/rfc5424`

[3] Mockapetris, P.: *Domain names - implementation and specification*. 1987, [Online].
URL `https://tools.ietf.org/html/rfc1035`

[4] Nyffenegger, R.: 2017, [Online].
URL
`https://renenyffenegger.ch/notes/development/Base64/Encoding-and-decoding-base-64-with-cpp`