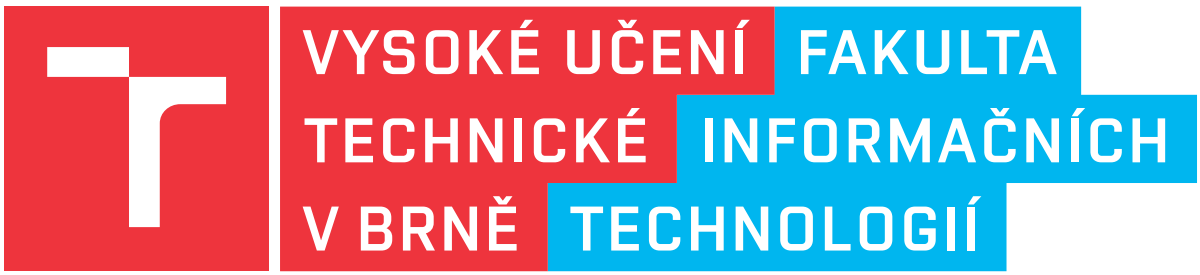
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Síťové aplikace a správa sítí

Export DNS informací pomocí protokolu Syslog

14. října 2018

Vladan Kudláč

Obsah

1 Úvod

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně. Dokumentace a vestavěná nápověda je psána v češtině. Manuálová stránka je stejně jako samotný kód psána v angličtině.

Cílem projektu bylo vytvořit aplikaci, která bude umět zpracovávat data protokolu DNS (Domain Name System) a vybrané statistiky exportovat pomocí protokolu Syslog na centrální logovací server.

2 Návrh a implementace

2.1 Zpracovávání paketů

Pro úspěšné zpracování DNS paketu je nutné pochopit princip knihovny pcap. Knihovna nabízí funkce pro zachytávání provozu ze síťového rozhraní i pro čtení ze souboru. Načtený paket je uložen v hexadecimální podobě v poli 8b hodnot. Při studování principu načítání, filtrování a získávání paketů jsem používal manuálové stránky na internetových portálech `https://liw.fi/manpages`, `http://www.cplusplus.com` a `http://man7.org`. Tyto stránky jsem využíval kdykoliv, kdy jsem potřeboval zjistit správné užití funkcí.

Služba DNS je protokol aplikační vrstvy [4] a je tedy nutné nejprve zpracovat ethernetovou, internetovou a UDP hlavičku. V ethernetové hlavičce zjistím verzi IP (4 nebo 6) a poté skočím na začátek DNS dat.

DNS obsahuje hlavičku a seznamy s dotazy a odpověďmi, viz RFC1035 [7]. V DNS hlavičce zjišťuji v příznacích zda se jedná o dotaz nebo odpověď. Dotazy zahazuji. Poté zjistím počet položek v sekcích `Questions` a `RRs answer`. Sekci `Questions` je nutné projít, aby jsme se dostali k sekci `RRs answer`.

Sekci `RRs answer` zpracovávám po jednotlivých položkách, vzájemně se neovlivňují. Nejprve zpracuji plošku `NAME`, kterou využívám ve statistikách pro **domain-name**. Název zpracovávám rekurzivně, aplikace si poradí i s vícenásobným odkazem. Hned poté následuje hodnota `TYPE`, která slouží pro druhou hodnotu **rr-type**. Aplikace rozpoznává celkem 42 typů [11], ale jen některé zpracovává. Poté přeskočím položky `CLASS` a `TTL` a uložím si hodnotu `RDLENGTH`. Následují samotná data formátovaná dle daného typu. V tuto chvíli se aplikace podívá na typ DNS záznamu, a pokud se nejedná o jeden z následujících typů: A, MX, NS, CNAME, SOA, TXT, AAAA, DNSKEY, RRSIG, NSEC nebo DS, tak s pomocí `RDLENGTH` skočí na další záznam.

Data pro **rr-answer** jsou formátována stejně, jako v programu `dig`. Pokud je zobrazeno více hodnot, jsou tyto hodnoty obaleny do uvozovek. Pro konkrétní formát vyhledejte manuálovou stránku programu `dig`.

Záznamy typu DNSKEY, RRSIG, NSEC a DS jsou součástí DNSSEC, který je implementován na základě informací v RFC4034 [9]. Formáty DNSKEY a RRSIG vyžadovaly implementovat vlastní base64 enkodér. Enkodér byl implementován na základě RFC4648 [5]. Dalším obzvlášť komplikovaným typem je `NSEC`, které obsahuje bitové pole s kompresí hodnot. Při implementaci jsem sbíral údaje z RFC4034 a z knihy „TCP/IP Illustrated, Volume 1: The Protocols“ [2].

2.2 Ukládání statistik

Pro ukládání statistik používám globální proměnnou, která je instancí objektu Stats. V projektu jsem si vyzkoušel objektové programování v C++ na základě informací z portálu Geeksforgeeks.org [6]. Objekt Stats má funkci `add` pro přidání záznamu do statistik, `print` pro výpis na standardní výstup a `send` pro odeslání na Syslog server. Hodnoty statistik jsou ukládány v poli jako struktura s položkami:

```
typedef struct dns_response {
        string domainName;
        string rrType;
        string rrAnswer;
        unsigned int count;
} dns_response;
```

2.3 Odesílání statistik

V případě, že není zadán server pro odesílání statistik, jsou na konci zpracování pcap souboru vytištěny statistiky funkcí `print` objektu Stats. Pokud se zpracovávají pakety z rozhraní, nic se nevypisuje.

Hned po spuštění se zaregistruje obsluha signálu `SIGUSR1`. Při příchodu signálu `SIGUSR1` dojde k paralelnímu vykonaní funkce, která vytiskne statistiku na standardní výstup. Při implementaci jsem se inspiroval článkem na Geeksforgeeks.org [8].

Pokud byl zadán server pro odesílání statistik a zároveň se pakety získávají ze síťového rozhraní, dojde před zahájením zpracování paketů k vytvoření druhého vlákna [10], které střídavě čeká po zadanou dobu a poté odesílá statistiky na Syslog server. Při čtení ze souboru jsou statistiky odeslány po zpracování souboru.

Příjem Syslog zpráv jsem zkoumal pomocí programu Wireshark a také nakonfigurováním RSYSLOG serveru ve virtuálním stroji CentOS7. RSYSLOG jsem dokázal nakonfigurovat dle návodu na portálu Tecmint [1]. Zprávy zasílám pomocí UDP. Každá statistika je v samostatné zprávě. Formát Syslog zpráv vychází ze zadání a ze standardu RFC5424 [3].

3 Uživatelská příručka

Projekt ISA: Export DNS informací pomocí protokolu Syslog. Verze 1.0 (19. 11. 2018).
Aplikace zpracovává DNS (Domain Name System) pakety z pcap souboru nebo získané odchytáváním komunikace na síťovém rozhraní a vytváří z nich agregované statistiky, které buď tiskne na standardní výstup, nebo je zasílá na Syslog server. Program umí zpracovávat následující typu DNS záznamů: A, MX, NS, CNAME, SOA, TXT, AAAA, DNSKEY, RRSIG, NSEC, DS. Jakýkoliv jiný záznam není započten do statistiky.

3.1 Požadavky

Program lze přeložit v překladači podporující standard *C++11*. Doporučuje se překladač ***gcc* verze 4.8.5 a novější**. Pro starší verze nebyl program testován. Program byl testován na následujících konfiguracích:

- GCC 7.3.0 (Ubuntu x86_64-linux-gnu) – Merlin + lokální vývoj
- GCC 4.8.5 (CentOS x86_64-redhat-linux) – Virtuální stroj

Knihovny potřebné k překladu

- pcap/pcap.h (nemusí být součástí běžných distribucí)
- iostream
- sstream
- string.h
- unistd.h
- netinet/if_ether.h
- netinet/ip.h
- netinet/udp.h
- arpa/inet.h
- sys/types.h
- sys/socket.h

- signal.h
- netdb.h
- vector
- ctime
- time.h
- thread

3.2 Překlad

Překlad lze provést programem *make*. Pro přeložení zadejte příkaz *make all* v adresáři s projektem. Pokud není možné použít program *make*, lze programy přeložit následujícím příkazem:

```
g++ -std=c++11 -static -libstdc++ main.cpp stats.cpp -o dns-export -lpcap -pthread
```

3.3 Spuštění

./dns-export [-r FILE] [-i INTERFACE] [-s SERVER] [-t INTERVAL]

- `-r FILE` – Zpracuje pakety ze zadaného `pcap` souboru. Po dokončení tiskne statistiku na standardní výstup nebo ji zasílá na Syslog server (v případě použití parametru SERVER). Nelze použít společně s parametry -r nebo -t.
- `-i INTERFACE` – Pakety budou zachytávány ze zadaného rozhraní. Zadejte „any“ pro zachytávání všech rozhraní. Program zasílá v časovém intervalu statistiky na Syslog server dokud není ukončen klávesou `CTRL+C`. Při obdržení signálu `SIGUSR1` tiskne statistiku na standardní výstup. Nelze použít společně s parametrem -i
- `-s SERVER` – Adresa (jmenná, IPv4 nebo IPv6) Syslog serveru pro zasílání statistik. Statistiky jsou zasílány v časových intervalech nebo po zpracování souboru.
- `-t INTERVAL` – Časový interval (v sekundách) zasílání statistik na Syslog server. Výchozí hodnota 60 s. Lze použít pouze s přepínačem -s.

3.4 Chybové kódy

- 0 – program skončil v pořádku
- 1 – chyba při zpracování argumentu
- 2 – chyba práce se soubory (vstupně/výstupní chyba)
- 3 – systémová chyba (chyba filtrování paketů)
- 4 – chyba síťového rozhraní (např. nelze vytvořit soket)

4 Závěr

Program byl testován na referenčním virtuálním stroji pro předmět ISA a na serveru Merlin. Kromě chybějící podpory pro pakety skrze UDP není znám žádný problém nebo omezení. Protokol UDP jsem neimplementoval, neboť jsem přesáhl hranici 40 hodin čisté práce, kterou jsem si pro tento projekt stanovil. Při implementaci jsem se opíral zejména o specifikace a oficiální dokumentaci.

Reference

[1] Cezar, M.: How to Create a Centralized Log Server with Rsyslog in CentOS/RHEL 7. Dostupné z: https://www.tecmint.com/create-centralized-log-server-with-rsyslog-in-centos-7, 2017, [online] [vid. 2018-11-16].

[2] Fall, K.; Stevens, W.: *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley Professional Computing Series, Pearson Education, 2011, ISBN 9780132808187.

[3] Gerhards, R.: The Syslog Protocol. RFC 5424, RFC Editor, 2008.
URL https://tools.ietf.org/html/rfc5424

[4] Ing. Petr Matoušek, M., Ph.D.: Systém DNS.

[5] Josefsson, S.: The Base16, Base32, and Base64 Data Encodings. RFC 4648, RFC Editor, 2006.
URL https://tools.ietf.org/html/rfc4648

[6] Kariya, A.: C++ Classes and Objects. Dostupné z: https://www.geeksforgeeks.org/c-classes-and-objects, 2018, [online] [vid. 2018-11-12].

[7] Mockapetris, P.: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. RFC 1035, RFC Editor, 1987.
URL https://tools.ietf.org/html/rfc1035

[8] Patel, K.: Signals in C language. Dostupné z: https://www.geeksforgeeks.org/signals-c-language, 2018, [online] [vid. 2018-11-16].

[9] R. Arends, M. L. D. M. S. R., R. Austein: Resource Records for the DNS Security Extensions. RFC 4034, RFC Editor, 2005.
URL https://tools.ietf.org/html/rfc4034

[10] Sayan Mahapatra, D. H.: Multithreading in C++. Dostupné z: https://www.geeksforgeeks.org/multithreading-in-cpp, 2018, [online] [vid. 2018-11-16].

[11] Wikipedia, P.: List of DNS record types. Dostupné z: https://en.wikipedia.org/w/index.php?oldid=488747971, 2012, [online] [vid. 2018-11-14].