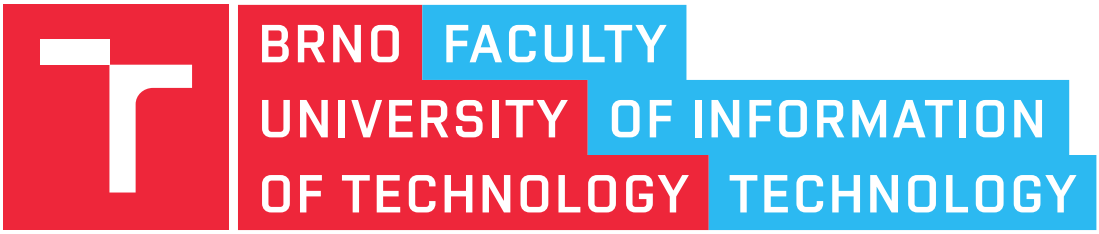


Síťové aplikace a správa sítí

Dokumentace k projektu Programování síťové služby

Varianta 2: Export DNS informací pomocí protokolu Syslog

15. listopadu 2018 Vladimír Dušek, xdusek27

Obsah

1 Úvod

Dokumentace popisuje řešení projektu a vysvětluje danou problematiku. Naší úlohou bylo nastudovat si potřebné informace a následně navrhnout, naprogramovat a otestovat síťovou službu, navíc k ní napsat manuálovou stránku. Varianta 2, kterou jsem řešil spočívala ve vytvoření aplikace dns-export. Ta odposlouchává síťový provoz na síťovém rozhraní, případně zpracovává pcap soubor, ve kterém je nějaká síťová komunikace již zaznamenána. Aplikace vyfiltruje DNS provoz a následně zpracovává jednotlivé pakety, konkrétně DNS odpovědi (responses). V každé odpovědi projde skrz všechny odpovědní záznamy (Answer resource records) a vyčte určité informace. Doménové jméno, typ DNS záznamu a data specifická pro každý typ. Dále je zaznamenán výskyt těchto informací, shodné jsou sečteny. Tyto výsledky jsou zasílany na syslog server ve formátu odpovídající syslog protokolu.

2 Uvedení do problematiky

2.1 Model TCP/IP

Síťová komunikace je kvůli své komplexnosti rozdělena do tzv. vrstev, které znázorňují hierarchii činností. Výměna informací mezi vrstvami je přesně definována. Každá vrstva využívá služeb vrstvy nižší a poskytuje své služby vrstvě vyšší.

2.1.1 Vrstva síťového rozhraní

Nejnižší vrstva, umožňuje přístup k fyzickému médiu. Je specifická pro každou síť v závislosti na její implementaci. (např. Ethernet)

2.1.2 Síťová vrstva

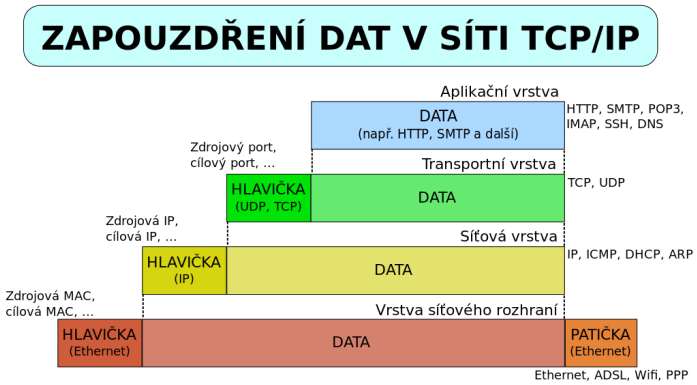
Vrstva zajišťuje síťovou adresaci, směrování a předávání datagramů. Je implementována ve všech prvcích sítě – směrovačích i koncových zařízeních. (např. IPv4, IPv6, ARP, ICMP)

2.1.3 Transportní vrstva

Poskytuje transportní služby pro kontrolu celistvosti dat. Jedná se o spolehlivé spojení (TCP) nebo nespolehlivé spojení (UDP). Je implementována až v koncových zařízeních, proto umožňuje přizpůsobit chování sítě potřebám aplikace.

2.1.4 Aplikační vrstva

Vrstva, která se stará o přenos konkrétních aplikačních dat. (např. SSH, FTP, HTTP, DHCP, DNS)



Obrázek 1: Schéma zapouzdření dat na vrstvách TCP/IP

2.2 Domain Name System

DNS (Domain Name System) je hierarchický systém doménových jmen, který je realizován servery DNS a protokolem stejného jména, kterým si vyměňují informace. Jeho hlavním úkolem jsou vzájemné převody doménových jmen a IP adres sítě. Slouží de facto jako distribuovaná databáze síťových informací.

2.2.1 Formát DNS zprávy

DNS zpráva má následující formát:

Header
Question
Answer
Authority
Additional

Header (hlavička) blíže specifikuje DNS zprávu. Například zda se jedná o dotaz, či odpověď, zda nastala nějaká chyba, nebo kolik zpráva obsahuje DNS záznamů (resource records) a jakých jsou typů.

V sekci Question (dotaz) jsou informace které chce tázající zjistit. Konkrétně doménové jméno na které se dotazuje, typ a třída záznamu.

Answer, Authority a Additional jsou pak záznamy. Všechny mají shodný formát a obsahují různé odpovědi, případně cestu jak se odpovědí dosáhlo.

2.3 Syslog protokol

Syslog protokol je standard pro záznam programových zpráv (logů). Program podle protokolu posílá zprávy napříč sítí ke kolektorům logovacích zpráv – syslog serverům. Jedná se tedy o architekturu klient-server. Komunikace může probíhat přes UDP na portu 514 nebo přes TCP na portu 6541.

2.3.1 Formát zpráv

Maximální délka paketu je 1024 bytů. Zpráva má následující formát:

PRI HEADER MSG

Část PRI se skládá ze třech až pěti znaků. Začíná znakem '<', následuje číslo a končí znakem '>'. Číslo uvnitř špičatých závorek se nazývá Priority value a reprezentuje Facility – zařazení podsystému a Severity – míru závažnosti.

PRI = <Priority Value>

Priority value se vypočítá následujícím vzorcem:

Priority Value = Facility * 8 + Severity

Část HEADER obsahuje časovou značku – TIMESTAMP, tedy kdy byla zpráva odeslána a adresu odesílatele – HOSTNAME, tedy kdo zprávu odeslal.

HEADER = TIMESTAMP HOSTNAME

Poslední část, MSG, se sestává opět ze dvou částí. První z nich je TAG, což je název procesu, který zprávu vygeneroval a druhá je CONTENT, tedy konkrétní obsah zprávy.

MSG = TAG CONTENT

3 Implementace

3.1 Struktura programu

Aplikace se snaží držet objektově orientovaného paradigma. Kvůli spravování signálů (`signal_handler`) bylo nutné některé objekty definovat na globální úrovni.

Jako první se pracuje s instancí třídy `ArgParser`, která se stará o zpracování argumentů příkazové řádky a jejich validaci. Argumenty jsou uloženy jako privátní atributy a přístup k nim je definován pomocí příslušných metod.

Třída `PcapParser` využívá `pcap` knihovnu pro odchytávání síťového provozu na síťovém zařízení, případně pro parsování zdrojového pcap souboru. Dále vyfiltruje DNS provoz a zpracuje veškeré hlavičky až k aplikační vrstvě. Poradí si na síťové vrstvě s IPv4 i IPv6 datagramama a na transportní vrstvě s TCP i UDP komunikací.

Třída `DnsParser` zpracovává data DNS protokolu. Vytřídí pouze odpovědi (responses), které neskončili chybou. Projde všechny záznamy v sekci Answers a vybere data potřebná k vytvoření statistik. Statistiky ukládá do globální datové struktury `unordered_map` jako dvojice klíč->hodnota. Klíč jsou získaná data z DNS záznamu a value je počet výskytů těchto dat za celou dobu odchytávání, případně za zpracování celého pcap souboru.

Třída `Syslog` se pak stará o komunikaci se syslog serverem. Poskytuje metody pro připojení, odpojení a odesílání zpráv na syslog server.

Ve funkci `signal_handler` jsou pak odchytávány signály. `SIGUSR1` pro výpis statistik na `STDOUT`, `SIGALRM` pro odeslání statistik na syslog server a `SIGINT` pro ukončení aplikace (v případě odposlouchávání na síťovém rozhraní běží aplikace až do obdržení tohoto signálu).

V modulu `utils` jsou implementovány pomocné funkce pro zpracovávání DNS komunikace, makro pro debug výpisy, nápověda, či výjimky pro zpracování chybného chování.

3.2 Popis implementace

Aplikace je napsaná v jazyce `C/C++` podle standardu `C++11`. Překlad je řešen pomocí Unixové utility `make` dle přiloženého `Makefile` souboru. Dále je napsána manuálová stránka ve značkovacím jazyce `troff` – soubor `dns-exp.1`, kde je stručný popis aplikace a jsou blíže specifikovány argumenty a návratové kódy.

Pro zpracování argumentů příkazové řádky je využívána knihovna `getopt.h`. Syntaxe zadávání argumentů programu, by tedy měla odpovídat standardním posixovým nástrojům. Například jsou povoleny zkrácené (`-a` i dlouhé (`--argument`) varianty argumentů a jejich libovolné pořadí.

Pro zachytávání provozu na síťovém rozhraní, či zpracovávání pcap souboru aplikace využívá funkce `pcap` knihovny. Ta také komunikaci vyfiltruje pouze na DNS komunikaci. Funkce `pcap_loop` iteruje přes veškeré vyfiltrované rámce. Každý z nich je zpracováván ve statické metodě `packet_handle`. Zde je zpracování dále děleno. Nejprve na vrtsvě síťového rozhraní (network interface layer) podle typu rámce – Ethernet, SSL. Poté na síťové vrstvě (network layer) podle typu datagramu – `IPv4`, `IPv6`. Nakonec i podle typu packetu na vrstvě transportní (transport layer) – `TCP`, `UDP`. Při `UDP` komunikaci je řízení rovnou předáváno modulu DnsParser. V případě TCP komunikaci, jsou data ukládána do globálního bufferu (`vector`) a zpracována až v případě nutnosti – před odesláním na syslog server, či vypsáním na `STDOUT`. Je tomu tak z důvodu fragmentace aplikačních dat DNS na transportní vrstvě.

Modul `DnsParser` pak sbírá už konkrétní informace. Tedy na jaké doménové jméno byl kladen dotaz, jaký typ dotazu byl požadován a konkrétní odpověď. Jsou podporovány následující DNS typy: A, AAAA, CNAME, DNSKEY, DS, MX, NS, NSEC, OPT, PTR, RRSIG, SOA, SPF a TXT. Ostaní typy jsou značeny jako `unknown_type` a jejich data jako `unknown_data`. Modul využívá různých implementovaných struktur pro ulehčení práce s ukazateli. U struktur je použit atribut `__attribute__((packed))`, aby nedocházelo k jejich zarovnání (padding).

Komunikace se syslog serverem je řešeno pomocí knihovny `sys/socket.h`. Nejprve je navázáno komunikace se serverem a poté jsou zprávy (logs) postupně odesílány. Před skončením aplikace je spojení ukončeno a socket uzavřen.

3.3 Návratové kódy

- 0 – OK
- 1 – chyba zpracování argumentů příkazové řádky
- 2 – chyba při odposlouchávání síťového rozhraní či zpracování pcap souboru (např. neplatné jméno rozhraní nebo pcap souboru)
- 3 – chyba při komunikaci se syslog serverem (např. nepodařilo se navázat spojení)
- 9 – systémová chyba (např. malloc nealokoval paměť)

4 Použití

4.1 Překlad

`$ make` – pro standardní překlad programu

`$ make pack` – pro překlad programu pro účely debugování (přepínač `-g`, debugovací vypisy)

`$ make clean` – pro smazání všech objektových souborů a deplistu

`$ make clean-all` – pro smazání všech objektových souborů, deplistu a binárního souboru

4.2 Spuštění

```
$ ./dns-export [-r RESOURCE | -i INTERFACE] [-t TIMEOUT] -s SERVER [-h]
```

- `-r RESOURCE`, `--resource RESOURCE` – Udává jméno pcap souboru, který se bude zpracovávát. Aspoň jeden z argumentů `resource` nebo `interface` musí být zadán.
- `-i INTERFACE`, `--interface INTERFACE` – Udává jméno rozhraní na kterém bude odposloucháván síťový provoz. Aspoň jeden z argumentů `resource` nebo `interface` musí být zadán.
- `-t TIMEOUT`, `--timeout TIMEOUT` – Časový interval v sekundách udávající, jak často statistiky budou zasílány. Výchozí hodnota 60 vteřin. Volitelný argument.
- `-s SERVER`, `--server SERVER` – Adresa syslog serveru kam statistiky budou zasílány. Povinný argument.
- `-h`, `--help` – Vypíše nápovědu na `STDOUT`. Volitelný argument.

4.3 Příklady spuštění

4.3.1 Zpracovávání pcap souboru

Spuštění aplikace pro zpracování `dns.pcap` souboru a odesílání zpráv (logs) na lokální syslog server.

```
$ ./dns-export -r dns.pcap -s localhost
```

Po ukončení běhu aplikace si můžeme zobrazit výsledné zprávý. Ty jsou uloženy podle konfigurace syslogu. Výchozí cesta je `/var/log/messages`. Pro jejich zobrazení můžeme použít například nástroj `journalctl`, který slouží pro zobrazení systémových zpráv. Ten obsahuje řadu přepínačů pro vyfiltrování konkrétnějšího obsahu. Pokud máme dostupná práva `root`, můžeme si soubor přímo vypsat. Například utilita `tail` nám vypíše posledních n řádků souboru.

```
# tail -n 5 /var/log/messages
```

```
<134>1 2018-10-26T13:57:28.703Z 192.168.122.1 dns-export --- www.youtube.com CNAME youtube-ui.l.google.com 1
<134>1 2018-10-26T13:57:28.703Z 192.168.122.1 dns-export --- widgets.getsitecontrol.com CNAME gscwidgets.b-cdn.net 1
<134>1 2018-10-26T13:57:28.703Z 192.168.122.1 dns-export --- youtube-ui.l.google.com A~172.217.23.238 1
<134>1 2018-10-26T13:57:28.703Z 192.168.122.1 dns-export --- www.yahoo.com CNAME atsv2-fp-shed.wg1.b.yahoo.com 1
<134>1 2018-10-26T13:57:28.703Z 192.168.122.1 dns-export --- photos-ugc.l.googleusercontent.com A~172.217.23.225 1
```

4.3.2 Odposlouchávání na síťovém rozhraní

Spuštění aplikace pro odposlouchávané provozu na síťovém rozhraní `eth0`. Jsou vyžadována práva `root`.

```
# ./dns-export -i eth0 -s localhost -t 30
```

Při obdržení signálu `SIGUSR1` nám aplikace vypíše na `STDOUT` dosud nasbírané statistiky. Jelikož zasíláme signál aplikaci která běží s právama `root`, potřebujeme je i při zaslání signálu. Abychom mohli zaslat procesu signál, potřebujeme znát jeho ID. To zjistíme například:

```
$ ps aux | grep dns-export
```

```
root      6445  0.0  0.1 372536  9484 pts/1    S+   13:05   0:00 sudo ./dns-export -i eth0 -s localhost -t 30
```

Zaslání signálu pak vypadá následovně:

```
# kill -SIGUSR1 6445
```

```
i.stack.imgur.com.cdn.cloudflare.net A~104.16.30.34 1
i.stack.imgur.com CNAME i.stack.imgur.com.cdn.cloudflare.net 1
cdn.sstatic.net A~151.101.193.69 1
stackoverflow.com A~151.101.1.69 1
cdn.sstatic.net A~151.101.1.69 1
```

Aplikace běží až do obdržení signálu `SIGINT`, ten lze zaslat aplikaci klávesami `CTRL+C`. Ještě před skončením jsou zaslány sesbírané statistiky na syslog server. Zobrazit si je můžeme stejně jako v případě se zpracováním pcap souboru.

5 Závěr

Celý projekt se skládá z následujících zdrojových souborů C++: `main.cpp`, `arg_parser.cpp`, `arg_parser.h`, `pcap_parser.cpp`, `pcap_parser.h`, `dns_parser.cpp`, `dns_parser.h`, `syslog.cpp`, `syslog.h`, `utils.cpp`, `utils.h` a souboru pro překlad `Makefile`. To je celkem 2208 řádků kódu. Velikost výsledného binárního souboru `dns-export` je 182.3 KiB. Dále dokumentace v sázecím systému LaTeX: `manual.pdf` a manuálové stránky ve značkovacím jazyce `troff`: `dns-export.1`.

Práce na projektu byla bez pochyby velmi zajímavá a přínosná. Zdokonalil jsem se ve psaní v C++, poznal pcap knihovnu, na vlastní kůži viděl jak funguje TCP/IP zásobník, poměrně do detailů si nastudoval DNS, vyzkoušel si komunikaci se syslogem a napsal manuálovou stránku. Na druhou stranu musím zmínit, že časová náročnost projektu velmi přesahovala deklarovanu časovou zátěž v kartě předmětu. K čemu tu pak máme kreditový systém ECTS s cílem sjednocovat studijní zátěž vysokoškolského studia napříč EU. Takto přestřelená časová zátěž je pak pouze na úkor docházky na přednáškách, práce na projektech do jiných předmětů, či, nyní ve třetím ročníku, na bakalářské práci.

Zdroje

[1] IANA, Domain Name System Security (DNSSEC) Algorithm Numbers [online], 2017-03-10, https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml, [cit. 2018-10-27]

[2] IANA, Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms [online], 2012-04-13, https://www.iana.org/assignments/ds-rr-types/ds-rr-types.xhtml, [cit. 2018-10-27]

[3] IANA, Domain Name System (DNS) Parameters [online], 2018-09-17, https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml, [cit. 2018-10-27]

[4] Network Working Group, RFC 1035 - Domain names - implementation and specification [online], 1987-11, https://tools.ietf.org/html/rfc1035, [cit. 2018-10-27]

[5] Network Working Group, RFC 3164 - The BSD syslog Protocol [online], 2001-08, https://tools.ietf.org/html/rfc3164, [cit. 2018-10-27]

[6] Network Working Group, RFC 3339 - Date and Time on the Internet: Timestamps [online], 2002-07, https://tools.ietf.org/html/rfc3339, [cit. 2018-10-27]

[7] Network Working Group, RFC 3596 - DNS Extensions to Support IP Version 6 [online], 2003-10, https://tools.ietf.org/html/rfc3596, [cit. 2018-10-27]

[8] Network Working Group, RFC 4034 - Resource Records for the DNS Security Extensions [online], 2005-03, https://tools.ietf.org/html/rfc4034, [cit. 2018-10-27]

[9] Network Working Group, RFC 5424 - The Syslog Protocol [online], 2009-03, https://tools.ietf.org/html/rfc5424, [cit. 2018-10-27]

[10] Wikipedia contributors, Domain Name System [online], 2018-10-26, https://en.wikipedia.org/wiki/Domain_Name_System/, [cit. 2018-10-27]

[11] Wikipedia contributors, List of DNS record types [online], 2018-10-18, https://en.wikipedia.org/wiki/List_of_DNS_record_types/, [cit. 2018-10-27]

[12] Wikipedia contributors, Domain Name System Security Extensions [online], 2018-10-05, https://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions, [cit. 2018-10-27]

[13] Redhat, How to configure remote logging with rsyslog [online], 2018-08-29, https://access.redhat.com/solutions/54363, [cit. 2018-10-27]

[14] Keyboard Banger, DNS Message Format And Name Compression [online], 2015-09-25, http://www.keyboardbanger.com/dns-message-format-name-compression/, [cit. 2018-10-27]

[15] firewall.cx, DNS QUERY MESSAGE FORMAT [online], 2012, http://www.firewall.cx/networking-topics/protocols/domain-name-system-dns/160-protocols-dns-query.html, [cit. 2018-10-27]

[16] Michael Kerrisk, man7.org, linux/man-pages [online], 2018-10, http://man7.org/linux/man-pages/man3/, [cit. 2018-10-27]

[17] Lars Wirzenius, Writing manual pages [online], 2018-07, https://liw.fi/manpages/, [cit. 2018-10-27]

[18] The Tcpdump team, Pcap library manual page [online], 2018-10, https://www.tcpdump.org/manpages/, [cit. 2018-10-27]