VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ANALÝZA ZACHYCENÉHO DNS PROVOZU

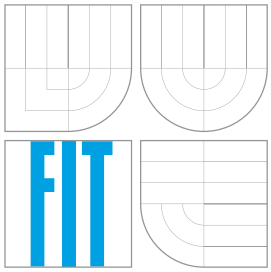
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JOZEF HMEĽÁR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ANALÝZA ZACHYCENÉHO DNS PROVOZU

ANALYSIS OF CAPTURED DNS TRAFFIC

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JOZEF HMEĽÁR

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL KOVÁČIK

BRNO 2015

Abstrakt

Táto práce je zaměřená na analýzu zachyceného DNS provozu. Úvod práce je zaměřený na základní popis počítačových sítí, službu DNS a popisu síťových toků. Pak se práce zaměřuje analýzou formátů Netflow, IPFIX a PCAP, analýzou a implementaci nástroje pro analýzu DNS provozu v jazyce C++. Závěr je věnovaný výsledkům implementovaného nástroje.

Abstract

This thesis is focused on the analysis of captured DNS traffic. Introduction of this thesis is focused of basic desciption of computer networks , DNS and description of network flows. Then, the work focused on analysis Netflow format, IPFIX and PCAP, the analysis and implementation of tool for analyzing DNS traffic in C++ programming language. The conclusion is devoted to the results of the implemented tools.

Klíčová slova

služba DNS, síťové toky, monitoring sítí, Netflow, IPFIX, PCAP, analýza provozu

Keywords

DNS, network flows, network monitoring, Netflow, IPFIX, PCAP, traffic analysis

Citace

Jozef Hmeľár: Analýza zachyceného DNS provozu, bakalářská práce, Brno, FIT VUT v Brně, 2015

Analýza zachyceného DNS provozu

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Michala Kováčika. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Jozef Hmeľár
20. května 2015

Poděkování

Chcel by som poďakovať vedúcemu mojej práce Ing. Michalovi Kováčikovi za odborné vedenie, pomoc a rady pri spracovaní tejto práce.

Obsah

Kapitola 1

Úvod

Počítačová komunikácia sa stala každodennou súčasťou bežného života. Sieťové služby využíva stále viac používateľov. Začiatkom deväťdesiatych rokov 20. storočia používala webové stránky len hŕstka nadšencov, ktorí vytvárili kód v HTML[1] v jednoduchých textových editoroch [10].

Trendom dnešnej doby je, že všetko chceme mať online. Ľudia si vytvárajú rôzne online účty na sociálnych sieťach (Facebook, Twitter, Google+, Instagram), využívajú rôzne služby ako YouTube, Gmail a pod. Nevlastniť účet na sociálnej sieti, nemať emailovú adresu, jednoducho nevyužívať niektorú z moderných online služieb je pre niektorých ľudí nepredstaviteľná záležitosť.

Tento stály rast počítačovej komunikácie kladie dôraz na efektívne využívanie zdrojov. Netýka sa to len nárokov na výkon procesora serverov a užívateľských staníc ale aj zariadení, ktoré zabezpečujú konektivitu a spoľahlivý prenos dát. Používateľ očakáva rýchle a spoľahlivé pripojenie. To zabezpečujú poskytovatelia internetového pripojenia a sieťoví administrátory. Tí potrebujú monitorovacie nástroje, aby čo najlepšie dokázali prispôsobiť požiadavky siete pre jej bezchybný chod.

Účelom monitorovania je zistiť výkon siete a prispôsobiť ho pre budúce požiadavky siete. Tento proces musí trvať dostatočne dlho, aby bolo možné zostaviť model správania siete. Dôležitou vecou pri monitoring je vybrať to, čo sa bude merať. Existuje veľké množstvo merateľných položiek [22].

Monitoranie prebieha aj za účelom sledovania anomálií na sieti. Cieľom detekcie anomálií je identifikovať prípady, ktoré sú v dátovom prenose neobvyklé, ide o akékoľvek výchylky z homogenného stavu. Autori [1] tvrdia, že distribúcie paketotých charakteristík, ktoré sa nachádzajú v zachytených tokoch, veľmi dobre ukazujú prítomnosť sieťových anomálií. S odhaľovaním sieťových anomálií je spojené aj odhaľovanie rôznych hrozieb. Typicky sa jedná o rôzne sieťové útoky, ktorých účelom je zahltiť sieť, zneprístupniť niektoré služby a podobne.

V článku [5] sa uvádza, že pre monitorovanie sieťovej prevádzky existujú aj ďalšie dôvody. Napríklad, charakteristiky vyťaženia siete výrazne ovplyvňujú sieťové komponenty a protokoly. Meranie sieťových charakteristík je dôležité aj v prípade vedecko-výskumnej činnosti. Výskumy sa väčšinou zameriavajú na získavanie znalostí o dynamike sietí. Pochopenie dynamiky sieťovej prevádzky je nevyhnutné z pohľadu budovania rôznych sieťových modelov, pre účely riešenia problémov týkajúcich sa vyhodnocovania, výkonnosti, zabezpečenia alebo optimalizácie sietí.

[1] HyperText Markup Language

Táto práca sa zaoberá monitorovaním a analýzou zachytenej Domain Name Server (DNS) prevádzky. V nasledujúcej kapitole 2 sa budem zaoberať teoretickou časťou práce, popíšem základné informácie o počítačových sieťach, o systéme DNS a o DNS paketoch. V kapitole 3 sa zaoberám analýzou a v kapitole 4 implementáciou daného nástroja. Kapitole 5 sa venuje testovaniu implementovaného nástroja z pohľadu časovej a pamäťovej náročnosti.

Kapitola 2

Úvod k problematike

V tejto časti priblížím zakládné informácie o sieťach a technológiach, ktoré sú v dnešnej dobe používané. Ďalej popíšem systém DNS, jeho dôležitosť pri sieťovej komunikácií. Priblížime si problematiku sieťových tokov, Pozrieme sa na nástroj `nfdump`, pre prácu s `Netflow`, knižnicu `Libpcap`, pre prácu s `PCAP súborom` a v poslednom rade si povieme niečo o nástroji pre tvorbu grafov `Gnuplot`.

2.1 Počítačové siete

Za posledné desaťročia pretrváva neustály nárast objemu a komplexnosti prenášaných infomácií vo forme dát. S rastúcim počtom počítačov súvisia aj väčšie požiadavky na zdieľanie a výmenu dát. Zdieľať môžeme informácie vo forme súborov, programov ale aj rôznych zdrojov (napr. hardverové zariadenia). Tieto požiadavky dali podnet pre vzniku prvých počítačových sietí.

V šesťdesiatych rokoch 20. storočia sa Ministerstvo obrany USA snažilo vymyslieť nový spôsob komunikácie. Do tohto programu boli zapojené aj popredné americké univerzity a to University of California a Massachusetts Institute of Technology. Ich spoločné snaženie nakoniec viedlo k vytvoreniu siete ARPA-net v roku 1968. Hlavnou myšlienkou bolo vytvoriť decentralizovanú počítačovú sieť, ktorá by nemala jeden hlavný bod a dokázala by fungovať aj v prípade, že niektoré jej časti by boli zničené. V sedemdesiatych a osemdesiatych rokov sa sieť začala postupne rozrastať.

Masové rozšírenie internetu v deväťdesiatych rokoch spôsobil vznik služby WWW[1]. Došlo k tomu na európskej pôde, konkrétne v ženevskom jadrovom centre CERN[2]. Autori WWW, Tim Berners-Lee a Robert Cailliau, použili známy princíp hypertextu - súbory textov navzájom prepojených odkazmi a pridali k tomu komunikačný protokol HTTP[3]. V tejto časti som čerpal hlavne z [14].

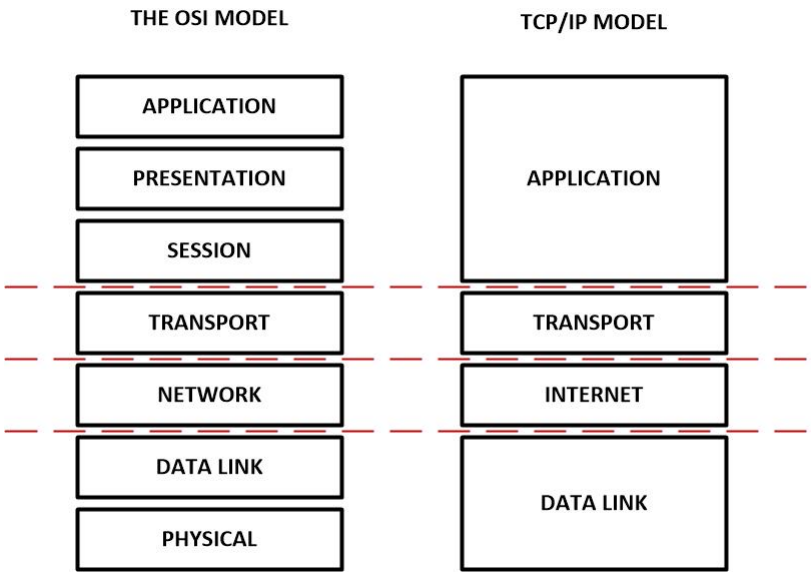
2.1.1 Modely sietí

V súčasnosti sa používajú dva modely architektúry počítačových sietí. Jeden z nich je ISO/OSI model (referenčný model) a druhý je TCP/IP model, ktorý je v dnešnej dobe štandardom internetu.

[1] World Wide Web
[2] anglicky European Organization for Nuclear Research
[3] Hyper-Text Transfer Protokol, RFC : http://tools.ietf.org/html/rfc2616

ISO/OSI model sa považuje za základ pre sieťové technológie. Tvorí ho sedem vrstiev 2.1, komunikovať je možné iba s vrstvou nad alebo pod. Každá z nich musí byť v komunikácií zainteresovaná, čo v rade praktický úloh prináša zbytočnú záťaž.

Z OSI/ISO modelu vychádza model TCP/IP. V súčasnosti tvorí základ komunikácie na internete. Tvoria ho štyri vrstvy, viď. obrázok 2.1. Pri odosielaní dát sa prevádza enkapsulácia (zapúzdrenie) od najvyššej vrstvy dole. Aplikačná vrstva vezme dáta, ktoré chce poslať inej stanici a doplní ich o aplikačnú hlavičku. Dáta pošle nižšej vrstve, transportnej, ktorá dáta rozdelí na segmenty, zabalí ich a pridá TCP[8] (alebo UDP[18]) hlavičku a vytvorí TCP segment. Ďalšia vrstva (sieťová) doplní IP hlavičku a takto vznike IP paket (IP datagram). V poslednej (prístupovej) vrstve sa k paketu pridá ethernetová hlavička na začiatok a trailer na koniec. Takto v poslednom kroku vznikne ethernetový rámec, ktorý sa vysiela na komunikačné médium. Keď cieľové zariadenie príjme dáta, prevádza sa opačný postup deenkapsulácia (rozbaľovanie) od najnižšej vrstvy hore a cieľová aplikácia dostane odosielané dáta. V tejto časti som čerpal najmä z [3].



Obrázek 2.1: Modely OSI/ISO a TCP/IP. Zdroj: [17].

2.2 Systém DNS

Systém DNS sa využíva na preklad doménového mena (zrozumieteľného názvu servera) na korenšpondujúcu IP adresu servera. Adresovanie a preklad adries patrí medzi dôležité súčasti internetovej komunikácie. Pri nefunkčnosti tejto služby nebude možné načítať žiadnu webovú stránku, či poslať mail.

Hlavnou úlohou služby DNS je mapovanie (preklad) doménových adries (napr. www.fit.vutbr.cz na IP adresu 147.229.9.23). Pre používateľov je vhodnejší zápis pomocou doménového mena (ľahšie zapamätanie textového reťazca, ako čísel).

Všetky aplikačné protokoly (napr. HTTP, SMTP[4], FTP[5]) používajú pre preklad doménových mien na IP adresy práve službu DNS. Každá akcia, ktorá si vyžaduje pripojenie

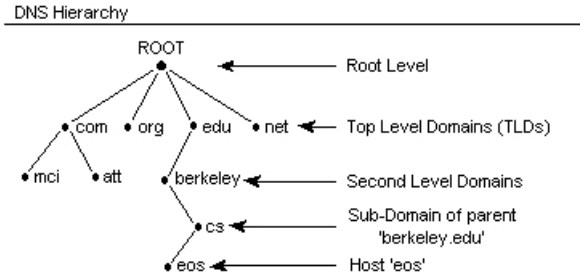
[4]Simple Mail Transfer Protocol
[5]File Transfer Protocol

na vzdialenú službu začína prekladom doménového mena. Najprv sa vyžiada preklad na IP adresu, potom sa môže nadviazať spojenie.

Systém DNS obsahuje databázu všetkých doménových mien a ich príslušných IP adries. Z toho vyplýva, že sa jedná o rozsiahlu databázu, ktorá je rozdistribuovaná na viacerých počítačoch, kde fungujú tzv. *nameservery*, doménové servery alebo tiež servery DNS.

Z dôvodu uloženia a efektívneho vyhľadávania je DNS hierarchicky usporiadané (logický priestor) ako koreňový strom doménových mien (viď. obrázok 2.2). Koreňu stromu DNS sa hovorí *the root*. Cesta od listu ku koreňu stromu je doménové meno.

Tento strom nie je uložený na jednom mieste, na jednom počítači. Jednotlivé časti podstromu celého priestoru doménových adries sú fyzicky uložené na lokálnych serveroch DNS, ktoré dohromady tvoria systém DNS. V tejto časti som čerpal informácie z [10].



Obrázek 2.2: DNS strom [7].

2.3 Rezolúcia požiadavku DNS

Rezolúcia (resolution, rozlíšenie) [10] je proces hľadania odpovedi v systéme DNS. Keďže je priestor doménových mien štruktúrovaný ako koreňový strom, stačí každému serveru DNS jediná informácia, ako vyhľadať ľubovoľný uzol v strome - adresa koreňového servera DNS. Server DNS sa opýta koreňového serveru DNS na najvyššiu doménu a potom postupuje od koreňa až k hľadanému uzlu, ktorý obsahuje hľadanú informáciu.

Koreňový server DNS (rootname server) je autoritatívny server pre všetky domény najvyššej úrovne (Top Level Domain). Pri zaslaní požiadavky na koreňový server, odpovie priamo hľadanou informáciou alebo vráti adresu DNS serveru, ktorý hľadanú informáciu obsahuje. Činnosť koreňového DNS servera popisuje RFC 2870 [2].

Koreňový DNS server je nepostrádateľný pre správny beh celého systému DNS. Preto existuje trinásť koreňových serverov [10] s adresami `[a-m].root-servers.net`, ktoré sú rozmiestnené po celom svete. Obvykle sa nejedná o jeden počítač, záťaž každého koreňového servera je rozložená na viacerých strojoch. Aj tak každý z nich odpovedá na tisíce žiadostí každú sekundu.

2.3.1 Resolver

`Resolver` je klientský program, ktorý zasiela požiadavky na dáta uložené v systéme DNS. Užívateľské programy, ktoré potrebujú informácie z DNS, pristupujú k týmto dátam pomocou resolveru.

Základné úlohy resolveru sú:

- Posielať žiadosti na servery DNS.

- Interpretovať odpovede od servera (prijaté záznamy, chýbové hlášky).
- Podať informácie užívateľskému programu, ktorý o informácie žiadal.

Resolver musí byť schopný pristupovať aspoň k jednému DNS serveru. Zo servera získa priamo hľadanú informáciu alebo mu server vráti odkaz na ďalší server, kde je hľadaná informácia uložená. Resolver tak môže preposielať požiadavky ďalším serverom.

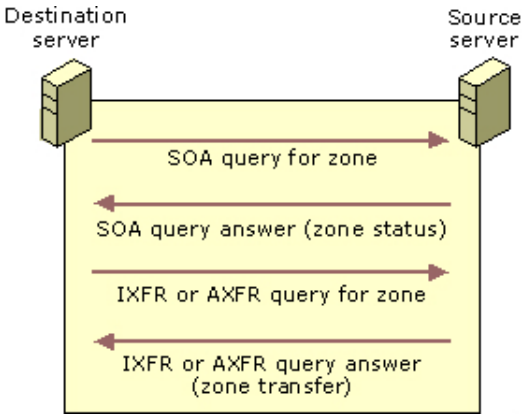
2.4 Zónový prenos

Jedná sa o prenos zónových súborov (všetky informácie týkajúce sa danej zóny) medzi primárnym (master) a sekundárnym (slave) DNS serverom. Využíva sa technika tzv. vyzývania (polling) sekundárnym serverom. Interval vyzývania (refresh interval) je uvedený v SOA záznamoch. Ak vyprší doba platnosti záznamu, sekundárny server zisťuje zmeny na primárnom DNS serveri a tieto zmeny nahráva aj do svojej databázy.

RFC 1996[6] prináša mechanizmus zasielanie upozornení o zmenách formou dotazu DNS NOTIFY.

Keď primárny server zistí podľa sériového čísla, že sa zmenila zóna, zašle špeciálne oznámenie všetkým sekundárnym serverom pre túto zónu. Keď sekundárny server obdrží správu NOTIFY (obrázok 2.3) začne sa správať, ako keby došlo k uplynutiu platnosti záznamu a požiada o prenos celej zóny (AXFR). Keď je zóna príliš veľká požiada o prírastkový prenos (IXFR). Pri tomto prenose sekundárny server požiada primárny server, ktorú verziu zóny má a požiada o posielanie zmien. Primárny server nájde vo svojej databáze všetky zmeny a tie pošle sekundárnemu serveru. Ak nie je nájdeny záznam o zmenách, posiela sa celá zóna, ako keby primárny server dostal príkaz AXFR.

Prenos celého zónového súboru je citlivá záležitosť z pohľadu bezpečnosti. Keďže sa posielajú informácie o celej doméne, primárny server má nakonfigurované IP adresy sekundárnych serverov, ktoré môžu požiadať o zónový prenos, koľko serverov sa môže súčasne prihlásiť a podobne. Pre tento prenos sa doporučuje vytvoriť zabezpečené spojenie medzi servermi. V tejto časti som čerpal najmä z [10].



Obrázek 2.3: Schéma zónového prenosu. Zdroj: [11].

[6] https://www.ietf.org/rfc/rfc1996.txt

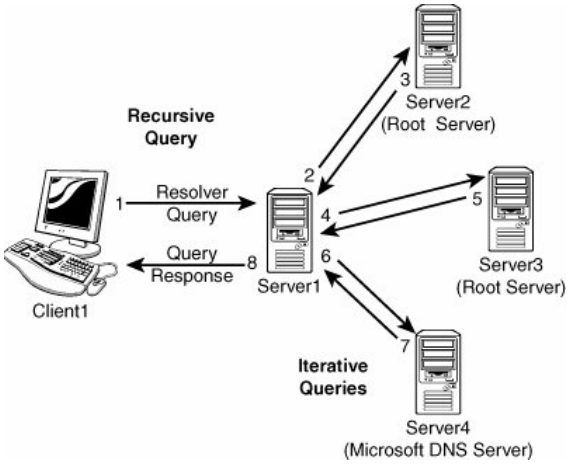
2.5 DNS Protokol

Vyššie sme si uviedli hlavné črty systému DNS. Teraz by sme sa bližšie pozreli nato, ako funguje DNS protokol. Bližšie si popíšeme akými spôsobmi dokáže resolver posielať žiadosti o predlad adresy.

Preklad adresy môžme popísať v týchto bodoch (za predpokladu, že je cache pamäť lokálneho DNS servera prázdna):

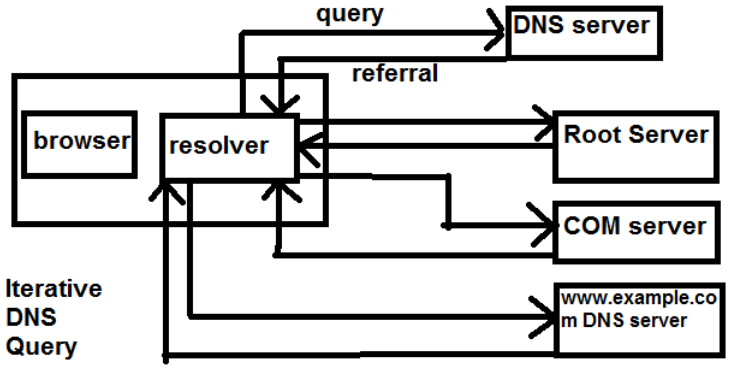
1. Klient požiada lokálny DNS server o preklad adresy.
2. Lokálny DNS server posiela požiadavok na koreňový DNS server o preklad adresy.
3. Koreňový DNS server odpovedá s odkazom na DNS server najvyššej úrovne (tj. sk, cz atď.).
4. Lokálny DNS server žiada DNS server najvyššej úrovne o predklad adresy.
5. Server najvyššej úrovne odpovedá s odkazom na sever druhej úrovne (napr. vutbr).
6. Lokálny DNS server žiada DNS server druhej úrovne o preklad adresy.
7. Ak tento DNS server druhej úrovne je autoritatívny DNS server pre požadovanú adresu, tak odpovedá IP adresou alebo chybou. Inak zasiela odkaz na DNS server tretej úrovne (napr. fit)
8. Lokálny DNS server zasiela klientovi odpoveď na jeho žiadosť.

Takémuto typu žiadosti sa hovorí rekurzívna žiadosť. Na obrázku 2.4 vidieť, ako to funguje. Takýto typ žiadostí využíva väčšina programov (napr. `dig, nslookup`.)



Obrázek 2.4: Sekvencia DNS správ: rekurzívna žiadosť. Zdroj: [13].

Ďalšou z možností posielania žiadostí je iteratívna žiadosť. Od tej rekurzívnej sa líši tým, že klient robí celý preklad adresy namiesto lokálneho DNS servera. Na obrázku 2.5 môžeme vidieť, ako táto iteratívna žiadosť vyzerá. Informácie použité v tejto časti som čerpal najmä z [19].



Obrázek 2.5: Sekvencia DNS správ: iteratívna žiadosť. Zdroj:[16].

2.6 DNS Pakety

DNS používa rovnaký formát paketu pre dotaz aj pre odpoveď. DNS paket pozostáva z dvoch, resp. piatich sekcií:

Header Hlavička DNS paketu. Obsahuje indentifikátory správy, ktoré generuje klient a server ich skopíruje do odpovede.

Question Sekcia dotaz. Nachádza sa v pakete dotazu a aj v pakete odpovede. Obsahuje doménové meno a ďalšie parametre.

Answer Sekcia odpoveď - na dotaz. Nesie záznamy, ktoré odpovedajú na dotaz.

Authority Sekcia obsahuje záznamy, ktoré popisujú ďalšie autoritatívne servery.

Additional Sekcia obsahuje záznamy, ktoré môžu byť užitočné pre záznamy z iných sekcií.

V každej správe je prítomná hlavička (Header). Tá obsahuje informácie, ktoré špecifikujú, ktoré z ostávajúcich sekcií sú prítomné. Ďalej špecifikuje, či ide o požiadavku (`query`) alebo odpoveď (`response`) alebo nejaký iní kód. Ďalej sa budem podrobnejšie zaoberať hlavičkou.

2.6.1 Hlavička DNS paketu

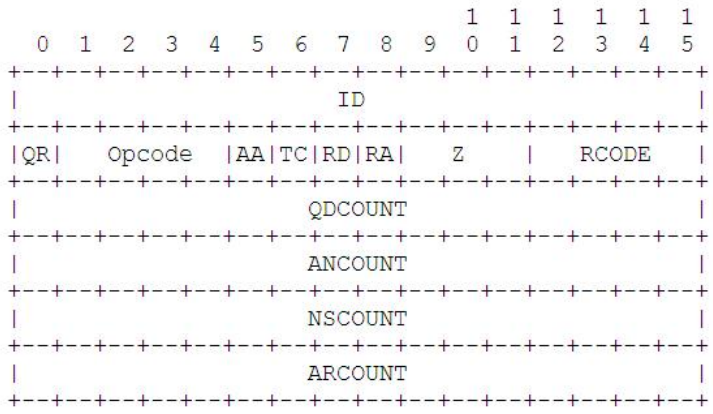
Hlavička DNS paketu 2.6, jednotlivé položky sú popísané nižšie.

ID

Šestnásť bitový identifikátor pridelený programom, ktorý generoval požiadavku.

QR

QR bit špecifikuje, či sa jedná o požiadavku (0) alebo o odpoveď (1).



Obrázek 2.6: DNS Packet Header. Zdroj: [12].

OPCODE

Štvor bitové pole, ktoré špecifikuje druh požiadavky. Túto hodnotu nastavuje odosielateľ požiadavky a kopíruje sa do odpovede. Môže nadobúdať tieto hodnoty:

(0)

Štandardná žiadosť (QUERY).

(1)

Inverzná žiadosť (IQUERY).

(2)

Žiadosť o stav servera (STATUS).

(3-15)

Rezervované pre budúce účely.

AA

Autoritatívna odpoveď - tento bit je platný v odpovediach a špecifikuje, že odpovedajúci DNS server je autoritatívnom serverom danej domény.

TC

Skrátenie (TrunCation) - špecifikuje, že táto správa bola skrátená vzhľadom k jej dĺžke, ktorá je väčšia ako je povolené na prenosovom kanále.

RD

Recursion Desired - bit môže byť nastavený v žiadosti a skopírovaný do odpovede. Ak je RD nastavené, DNS server sa usiluje o rekurzívne dotazy.

RA

Recursion Available - tento bit je nastavený alebo nulovaný v odpovedi. Značí, či sú rekurzívne dotazy podporované na DNS serveri.

Z

Rezervované pre budúce účely. Musí byť nulové vo všetkých dotazoch a odpovediach.

DNS response codes

Jednou z položiek protokolu DNS je response code (4 bitová hodnota, RCODE tj. kód odpovede). Keď server odpovedá na požiadavku nastavuje RCODE na príslušnú hodnota. Táto hodnota indikuje, či všetko prebehlo v poriadku alebo indikuje chybu a taktiež o akú chybu sa jedná. Návratové kódy môžu byť:

No error (0)

Žiadna chyba nenastala.

Format error (1)

DNS server nie je schopný interpretovať požiadavku.

Server failure (2)

Chyba na strane servera, nedokáže spracovať žiadosť.

Name error (3)

Iba pre odpovede (responses) z autoritatívneho servera, kód hovorí, že doménové meno v požiadavku neexistuje.

Not Implemented (4)

DNS server nepodporuje typ žiadosti.

Refused (5)

DNS server odmietol vykonať danú operáciu (nie je povolená).

Reserved (6-15)

Vyhradené pre budúce využitie.

QDCOUNT

Bezznamienkový šestnásť bitový integer špecifikuje počet položiek v sekcii QUESTION.

ANCOUNT

Bezznamienkový šestnásť bitový integer špecifikuje počet záznamov v sekcii ANSWER.

NSCOUNT

Bezznamienkový šestnásť bitový integer špecifikuje počet DNS serverov v sekcii AUTHORITY.

ARCOUNT

Bezznamienkový šestnásť bitový integer špecifikuje počet záznamov s sekcii ADDITIONAL.

V tejto časti som čerpal hlavne z [12]. V tabuľke 2.1 je prehľad šťandardov pre základné typy DNS záznamov.

Typ záznamu	Názov	Štandard
NS	Name Server	RFC 1034
A	Address	RFC 1034
MX	Mail Exchanger	RFC 1034
CNAME	Canonical Name	RFC 1034
PTR	Domain Name Pointer	RFC 1034
NAPTR	Naming Authority Pointer	RFC 2915,3403,3761
TXT	Text	RFC 1034
SRV	Service Record	RFC 2782
LOC	Location	RFC 1876
SOA	Start of Authority	RFC 1035, 2308
AAAA	IPv6 Address	RFC 3596
DNSKEY	DNS Key Record	RFC 4034
RRSIG	DNSSEC Signature	RFC 4034
NSEC	Next-Secure Record	RFC 4034
NSEC3	NSEC record version 3	RFC 5155
DS	Delegation Signer	RFC 4034

Tabulka 2.1: Prehľad štandardov pre základné typy DNS záznamov.

2.7 Sieťové toky

Medzi hlavné požiadavky súčasnej doby patrí vysoká dostupnosť služieb a bezpečnosť sietí. Tieto požiadavky vychádzajú z dnešných trendov, keď sa chod bežných aplikácií presúva na servery, vznikajú dátové uložiská, využíva sa virtualizácia a cloudcomputing. Stále stúpa množstvo prenášaných informácií. Najkritickejšou z požiadaviek je dostupnosť siete. Aby mal administrátor sieť pod kontrolou, musí mať prístup k monitoringu jednotlivých staníc v sieti, sledovaniu dátových tokov a údajom o vyťaženosti jednotlivých liniek. Na základe týchto údajov je schopný účinne spravovať sieť.

V dnešnej dobe je vhodným riešením používanie tzv. `flow-u dát`. Jedná sa o pakety zo siete združované do sieťových tokov, pričom tokom sa rozumie postupnosť paketov majúcich spoločnú vlastnosť a prechádzajúcich bodom pozorovania za určitý časový interval. Všetky pakety, ktoré patria do jedného toku majú spoločné vlastnosti odvodené z obsahu paketu. Tok je sekvencia paketov indentifikovaná podľa zdrojovej a cieľovej IP adresy, zdrojového a cieľového portu a protokolu za stanovený časový interval. Pri sieťových tokoch je možné rozlišovať smer komunikácie.

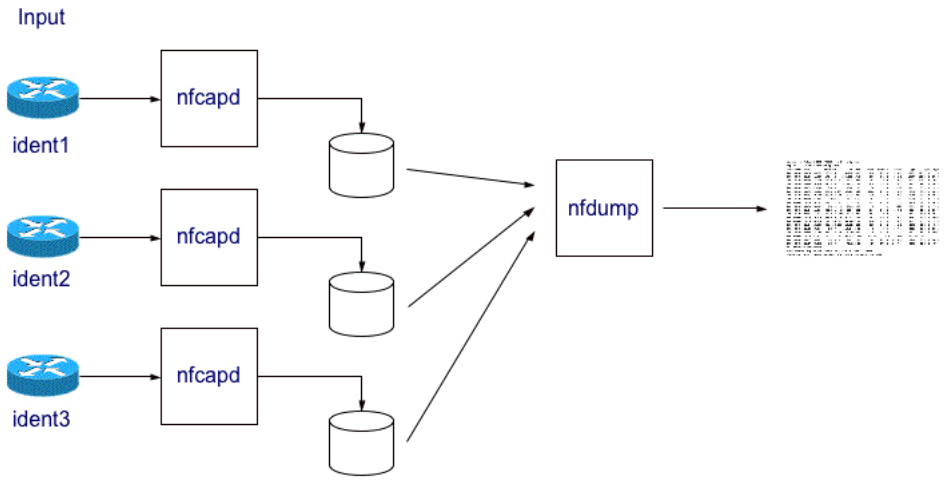
Vzhľadom k tomu, že sú uchovávané len určité údaje o tokoch, môže rýchlosť spracovania prekračovať až 10Gbps. Nevýhodou je však neprítomnosť ostatných údajov o toku a údajov z jeho paketov.

Na sledovanie tokov sú do siete nasadené `kolektory` a jeden alebo viac `exportérov`. Úlohou exportéra je odosielanie monitorovacích dát na kolektor. Túto úlohu vykonávajú smerovače alebo sa používajú špeciálne monitorovacie zariadenia tzv. `sondy`. V práci budem využívať formáty, ktoré slúžia pre uchovanie sieťovej prevádzky a to Netflow, IPFIX[7] a PCAP[8]. V tejto časti som čerpal hlavne z [9].

2.7.1 NFDUMP

NFDUMP je program, ktorý zobrazuje a analyzuje Netflow dáta. Tento nástroj číta Netflow zo súboru uloženého ako nfcapd a spracováva toky podľa zadaných parametrov. Je distribuovaný pod licenciou BSD[9]. Cieľom návrhu je schopnosť analyzovať zachytené Netflow dáta, rovnako ako nepretržite sledovať zaujímavé vzorky prevádzky. Čas uchovania Netflow dát je obmedzený iba na dostupné miesto na disku. Nástroj je optimalizovaný pre rýchle a efektívne filtrovanie.

Všetky zachytené dáta sú pred analýzou uložené na disk. To oddeľuje proces uloženie a analýzy dát. Na obrázku 2.7 je znázornené ako vyzerá zachytávanie dát. Kolektor za určitý časový interval (typicky 5 minút) prerotuje a premenuje výstupné súbory s časovou značkou a vytvorí súbory `nfcapd.YYYYMMddhhmm`, takže napríklad súbor `nfcapd.201505150900` bude obsahovať dáta z 15. mája 2015 09:00. Ak sa každých 5 minút bude vytvárať takýto súbor, za jeden deň sa ich vytvorí 288. V tejto časti som čerpal hlavne z [20].



Obrázek 2.7: Zachytávanie paketov. Zdroj: [20].

2.7.2 LIBPCAP

Libpcap [6] je open source knižnica, ktorá poskytuje rozhranie pre sieťové paketov. Bola vytvorená v roku 1994 výzkumníkmi z Lawrence Berkeley National Laboratory z University

[7] IP Flow Information Export
[8] Packet Capture
[9] http://www.linfo.org/bsdlicense.html

of California.

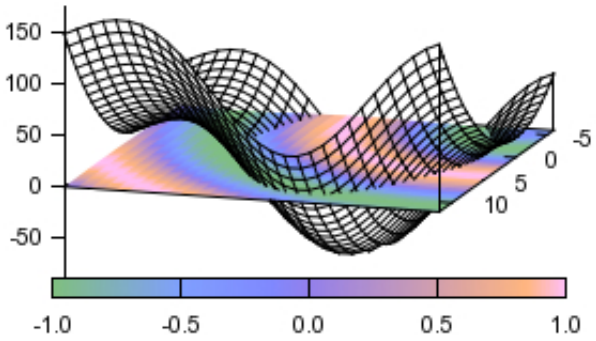
Každý predajca OS mal implementovaný vlastný mechanizmus zachytávanie paketov. Hlavným cieľom autorov bolo toto eliminovať a vytvoriť platfomovo nezávislé API. Libpcap je predurčený pre používanie v jazyku C alebo C++. Avšak, môžu ho využívať aj iné programovacie jazyky ako Perl, Python, Java alebo Ruby. Libpap beží na unixovo orientovaných operačných systémoch (Linux, Solaris, BSD). Existuje tiež verzia pre Windows, Winpcap.

V súčasnosti udržuje libpcap skupina Tcpdump[10].

2.8 Gnuplot

Gnuplot[21] je prenosná grafická utilita pre Linux, OS/2, MS Windows a ďalšie platformy. Je riadená cez príkazový riadok. Zdrojový kód je chránený autórskym právom, ale je voľne šíriteľný (nemusí za neho platiť).

Pôvodne bol vytvorený pre vedcov a študentov na vizualizáciu matematických funkcií 2.8 a interaktívnych údajov ale vzrástol na podporu mnoha neinteraktívnych použití, napríklad ako web scripting. Gnuplot je napríklad využívaná aplikáciou Octave[11]. Gnuplot je podporovaný a pod aktívnym vývojom od roku 1996.



Obrázek 2.8: Ukážka grafu z utility Gnuplot. Zdroj [21].

[10] www.tcpdump.org
[11] http://www.gnu.org/software/octave/

Kapitola 3

Analýza a návrh

V tejto kapitole priblížim zdroje dát, ktoré sú použité v tejto práci. Popíšem formáty Netflow, IPFIX a PCAP. Zhodnotím výber týchto zdrojov a poukážem na ich klady a zápory. Ďalej budem rozoberať vybrané položky, z ktorých sa budú vytvárať štatistiky. Koniec kapitoly bude venovaný základnemu návrhu aplikácie, aké požiadavky boli zohľadnené pri návrhu.

3.1 Zdroje dát

NetFlow[1] je otvorený protokol vytvorený spoločnosťou CISCO. Pôvodne bol určený ako doplnková služba pre CISCO smerovače. Pomocou neho môžme nahliadať do dát zo siete v reálnom čase.

Medzi základné prvky systému NetFlow patrí:

Exportér (sonda)

Zariadenie, ktoré zisťuje a ukladá si informácie o tokoch do dočasnej pamäte, ktorej sa tiež hovorí `NetFlow cache` a po čase ich odosiela na kolektor. Sleduje jednotlivé pakety a pozerá sa, aké majú jednotlivé pakety charakteristiky (zdrojova, cieľova IP adresa atď.). Ak sa podarí zachytiť takáto postupnosť paketov, exportér to nazve tokom a založí záznam pre daný tok.

Dáta na sondách môžu expirovať rôznymi spôsobmi. Ak sa zaplní NetFlow cache pamäť (staré neukončené záznamy sa začnú odstraňovať), v TCP spojení ak sa zachytí príznak RST alebo FIN. Tak sú tam ešte dva typy časovačov, jeden je aktívny (tok aktívny po dobu napr. 30 minút, potom končí, odošle sa do kolektora) alebo inaktívny (po nejakom časovom intervale, ak nepríde žiaden paket z daného toku, tok končí, odošle sa do kolektora).

Kolektor

Kolektor je zariadenie na sieti, ktoré funguje ako databáza, tzn. ukladá si dáta v špeciálnom formáte a komunikuje s viacerými sondami. Nad kolektorom sa vytvárajú dotazy a z dát sme schopný získať nazbierané informácie (kto s kým komunikoval, v akom čase, akým protokolom, nevidíme obsah komunikácie, ale vieme že prebehla).

[1] RFC 5101 http://tools.ietf.org/html/rfc5101, RFC 5102 http://tools.ietf.org/html/rfc5102

Protokol NetFlow počas svojej existencie prešiel niekoľkým verziami. Najrozšírenejšou sa stala verzia 5. Tok sa identifikuje na základe zdrojovej a cieľovej IP adresy, zdrojového cieľového portu, protokolu, rozhrania, na ktorom bol tok zachyteny a typom služby (Type of Service). Rozlišuje sa smer komunikácie. Tiež sa zaznamenávajú počty paketov pre tok, počet prenesených bajtov v toku, časové značky začiatku a konca toku a nastavené priznaky v prípade protokolu TCP.

NetFlow verzia 9 patrí medzi ďalšie významné verzie tohto protokolu. Hlavnou výhodou je šablonovanie, tzn. nie sú tam fixné položky, definujú sa šablony - zvolíme si, aké dáta budeme exportovať, aké sú veľké a čo znamenajú. Na exportér sa najprv pošle šablóna (dátová množina tokov) a potom sa posielajú dáta.

Verzia 9 ďalej dovoľuje preniesť okrem položiek z verzie 5 aj MPLS[2] toky alebo IPv6 adresy s portami. Protokol NetFlow bol vždy pod vedením firmy CISCO, nikdy nebol štandardom.

IPFIX Internet Protocol Flow Information eXport je narozdiel od NetFlow štandardizozaný protokol IETF (The Internet Engineering Task Force [3]). NetFlow verzia 9 je predchodca IPFIX, obsahuje však niekoľko rozšírení. Tak ako NetFlow nemá pevne danú štruktúru prenášaných dát. Dovoľuje definovať nové položky pre prenos, dovoľuje pridať vybrané informácie [15]. IPFIX podporuje premennú dĺžku prenášaných polí (NetFlow nepodporuje). Táto skutočnosť je užitočná pri prenose URL adries či doménových mien pri DNS.

3.1.1 Vhodnosť dát

Netflow dokáže monitorovať širší rozsah infomácií o paketoch a tým vieme získať nové infomácie o správaní siete. Inými slovami vieme špecifikovať, čo presne chceme zachytiť. Pre analýzu DNS prevádzky však potrebujeme dáta z aplikačnej vrstvy a tie Netflow nedokáže zachytiť. Zaujímavé položky, ktoré môžeme sledovať sú počty prenesoných paketov v toku, veľkosť prenesených paketov a podobne.

Pre dôkladnejšiu analýzu DNS prevádzky je vhodnejším formát IPFIX[4], ktorý je schopný zachytiť a exportovať viac položiek ako NetFlow.

Formát PCAP je pre analýzu DNS vhodný, pretože pakety obsahujú všetky položky, ktoré sú potrebné pre získanie potrebných výsledkov. PCAP je však veľmi pomalý na spracovanie.

3.1.2 CESNET2

Zdroje dát pre analýzu boli dodané zo siete CESNET2. Združenie CESNET[5] založili vysoké školy Akadémie vied Českej republiky v roku 1996. Jeho hlavným cieľom je výzkum a vývoj informačných a komunikačných technológií, budovanie a rozvoj e-infraštruktúry CESNET určenej pre výzkum a vzdelanie.

Základným prvkom celej e-infraštrukúry je vysokorýchlostná počítačová sieť CESNET2, ktorej chrbticová časť prepojuje okruhy s vysokými prenosovými rýchlosťami najväčšími univerzitnými mestami Českej republiky a ďalšími oblasťmi.

Jadro topologie siete CESNET2 je bohaté na DWDM infraštruktúru s desiatkami prenosových kanalóv o rýchlostiach 100, 10 a 1 Gb/s. Topológia siete sa skladá z kruhov

[2] Multiprotocol Label Switching
[3] http://www.ietf.org/
[4] http://www.iana.org/assignments/ipfix/ipfix.xml
[5] www.cesnet.cz

prechádzajúcich obmedzeným počtom miest. Cieľom je redudantná chbticová sieť s nie príliš dlhými trasami a taktiež s malými oneskoreniami, ktoré vznikajú na aktívnych prvkoch siete.

3.1.3 FlowMon exportér od INVEA-TECH

FlowMon tvorí komplexné riešenie pre monitorovanie sietí na báze tokov (NetFlow/IPFIX). Nás bude zaujímať konkrétne DNS plugin pre FlowMon exportér, ktorý dokáže parsovať DNS prevádzku a vybrať DNS dáta. Plugin rozširuje možnosti exportéra o parsovanie DNS dát a export dát do novo definovaných bodov využívajúcich formát IPFIX.

DNS plugin pre FlowMon exportér spracováva DNS prevádzku, vyberá niektoré položky z DNS paketov. Tieto položky sa nachádzajú na aplikačnej vrstve a obsahujú podstatné informácie pre analýzu DNS prevádzky. Exportované DNS položky sú vyberané s ohľadom na analýzu anomálií a detekciu útokov.

3.2 Použiteľnosť položiek

Monitorovanie siete sa používa hlavne na zistenie charakteristík siete. Z monitorovaných položiek dokáže administrátor zistiť o aký typ siete sa jedná, dokáže zistiť, ako je sieť vyťažená, aké veľké dáta sú po sieti prenášané, úspešnosť sieťovej komunikácie jednotlivých služieb a podobne. Monitoring by mal ďalej odhaliť prípadné sieťové anomálie, napríklad môže sa jednať o rôzne útoky na rôzne služby. Na základe týchto požiadaviek kladených na monitoring siete som sa rozhodol vytvárať štatistiky z týchto položiek zachytenej DNS prevádzky.

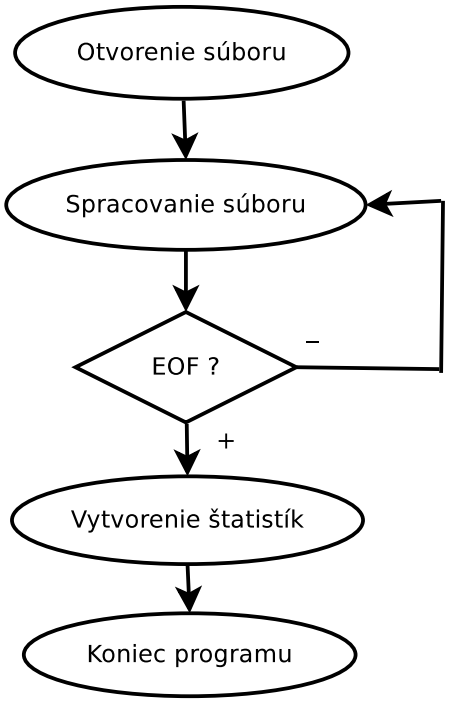
- Zistenie dĺžky doménových mien, výpočet primernej dĺžky doménové mena v súbore.
- Pomer návratových kódov RESPONSE paketov, zistíme úspešnosť dotazov, ktoré v súbore nájdeme.
- Zistenie typov dotazov, zastúpenie typov dotazov v súbore.
- Pomer typov odpovedi, zastúpenie odpovedí typov dotazov v súbore.
- Pomer dotazov a odpovedi, jedná sa o pomer medzi Questions a Answers položkami v paketoch.
- Pomer QUERY a RESPONSE paketov v súbore, koľko odišlo dotazov, koľko sa vrátilo odpovedí.
- Priemerná veľkosť DNS paketov, ako aj priemerná veľkosť QUERY a RESPONSE paketov.
- Čas platnosti jednotlivých odpovedí v súbore, položka TTL.
- Veľkosť položky RLENGTH v súbore.
- Pomer medzi IPv4 a IPv6 adresami v priloženom súbore.
- Zastúpenie portov v priloženom súbore.

3.3 Základný návrh aplikácie

Aplikácia by mala mať čo najgeneralizovanejšie použitie, hlavne keď sa jedná o aplikáciu, ktorá bude spracovávať veľké objemy dát (obzvlášť pri formátoch Netflow a IPFIX). Niektoré operácie pri spracovaní môžu byť časovo náročné a mohli by spomaľovať celý proces analýzy súboru. Preto je potrebné, aby aplikácia bola čo najviac parametrizovateľná, tzn. že používateľ si bude môcť vybrať z rady parametrov, podľa ktorých sa spracuje vstupný súbor.

Z vyššie uvedeného vyplýva, že chod aplikácie sa bude riadiť parametrami zadanými pred jej spustením. Po spustení aplikácie už nebude možné meniť jej parametre.

Na obrázku 3.1 je jednoduchý návrh aplikácie. Po vykonaní všetkých potrebných operácií, ktoré sú potrebné pre získanie užívateľom zvolených štatistík sa na štandardný výstup (STDOUT) vypíšu jednotlivé štatistiky. Ak sa bude jednať o štatistiku typu histogram dĺžky doménových mien, histogram obsadenie portov, histogram TTL hodnôt, tak výstupné dáta budú zapísané do súboru. Pre vytvorenie histogramov je nutné spustiť skript, ktorý daný graf vygeneruje.



Obrázek 3.1: Diagram aplikácie.

Kapitola 4

Implementácia

V tejto kapitole popíšem implemtáciu nástroja, ktorý bol analyzovaný v kapitole 3. Nástroj som sa rozhodol implementovať v jazyku C++. Výhodou je, že knižnica Libpcap popísanú bola vytvorená práve pre tento jazyk. Ďalšou výhodou tohto jazyka je, že poskytuje knižnice pre spracovanie Netflow dát. V iných programovacích jazykov je práca so súbormi Netflow komplikovanejšia.

4.1 Spracovanie dát

Spracovanie dát prebieha rôzne, závisí hlavne od vstupného súboru.

PCAP súbor je spracovaný knižnicou Libpcap. Dáta z aplikačnej vrstvy sú uložené v buffery, z ktorého sa parsujú potrebné položky. Algoritmus pre spracovanie paketu z tohto súboru sa dá popísať v týchto krokoch:

1.krok Otvorenie súboru.

2.krok Načítanie paketu do štruktúry. Jednotlivé vrstvý sú reprezentované v kóde ako štruktúry, ktoré obsahujú položky na danej vrstve.

3.krok Vyfiltrovanie paketov. DNS komunikuje na porte 53. Na transportnej vrstve sa filtrujú pakety podľa cieľového a zdrojového portu. Ak sa nejedná o DNS port, tak sa vrati na krok 2.

4.krok DNS dáta z odchyteného paketu sú nahrané do bufferu, s ktorým sa v ďalšej časti pracuje. Potrebné položky pre analýzu sa priebežne ukladajú. Po skončení analýzy sa opakuje krok 2. Ak sa sa v súbore nenachádzajú žiadne pakety, pokračuj na krok 5.

5. krok Zo zozbieraných údajov sa vytvoria požadované štatistiky. Program končí.

Použité knižnice a ukážka kódu pre spracovanie PCAP súboru 1.

Algoritmus 1: Spracovanie PCAP.

```
#include <stdio.h>
#include <pcap.h>
#include <stdlib.h>
#include <netinet/ip.h>
#include <arpa/inet.h> //použité knižnice pre PCAP
pcap_open_offline(file.c_str(), errbuff); //otvorenie súboru
while (pcap_next_ex(pcap, header, packet) >= 0) //načítanie packetov
{ //spracovanie packetu }
return 0;
```

CSV súbor je spracovaný pomocou knižnice `cstring`. Položky v tomto súbore sú uložené ako textové reťazce oddelené čiarkami[1]. Jeden riadok predstavuje jeden záznam o toku. Na začiatku súboru je hlavička, v ktorej sú vypísané položky, ktoré môžme v súbore nájsť. Jednotlivé položky sú oddelené čiarkami. Pozícia položky v hlavičke odpovedá pozícii položky v zázname. Pri prejdení celého csv súboru sa vytvoria požadované štatistiky a program skončí.

Netflow súbor je spracovaný podobne ako vyššie popísaný PCAP. Z algoritmu 1 a 2 vidíme, že spracovanie týchto súborov je podobné, len sú použité iné knižnice a funkcie. S tým rozdielom, že pri tomto type súboru sme schopný vytvoriť len obmedzené štatistiky.

Algoritmus 2: Spracovanie Netflow.

```
#include "libnfdump.h" //použité knižnice pre Netflow
nfdump_iter_start(infile_binary, fvalue, ) //otvorenie súboru
while(nfdump_iter_next(infile_binary, flowdata) != NFDUMP_EOF)
  //načítanie tokov
{ //spracovanie toku }
return 0;
```

Dáta získané z jednotlivých položiek sú úložené ako globálne premenné. Zväčša sa jedná o dátové typy typu `unsigned long long int` alebo `u_int`. Niektoré štatistiky si vyžadujú zachovať svoje údaje v poli. Napríklad dĺžky doménových mien sú uložené v poli, kde index na položku poľa udáva dĺžku získaného doménoveho mena a položka v poli, počet nájdených doménových mien s danou dĺžkou. Aby boli premenné viditeľné v celom programe, deklaroval som ich kľučovým slovom `extern`.

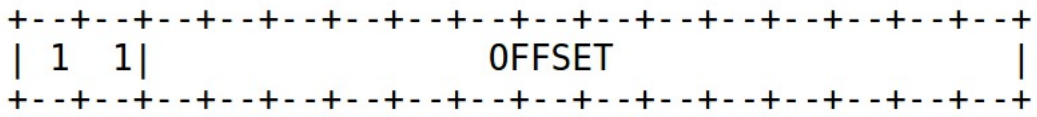
4.2 Kompresia doménových mien

Pre získanie dĺžky doménového mena z PCAP súboru bolo potrebné vyriešiť, ako sa dostať k celému názvu doménoveho mena.

Doménový systém využíva kompresiu doménových mien [12], ktorá má eliminovať opakovanie doménových mien v častiach NAME, QNAME a RDATA, za účelom zníženia veľ-

[1] Comma Separated Value

kosti správ. V tomto prípade, celé doménové meno alebo časť doménového mena je nahradená ukazovateľom na predchádzajúci výskyt rovnakého mena. Ukazovateľ sa skladá z dvoch octetov:



Obrázek 4.1: Pointer. Zdroj: [12].

Z obrázka 4.2 vidíme, že prvé dva bity sú jednotky. To umožňuje rozlíšiť ukazovateľ od značky (počet znakov doménového mena), ktoré musia začínať dvoma nulovými bitmi a sú obmedzené na 63 alebo menej.

Ukazovateľ špecifikuje pozíciu od začiatku DNS správy, tzn. prvý oktet ID poľa v hlavičke DNS paketu. Ukazovateľ 0 bude znamenať prvý oktet poľa ID, atď.

Režim kompresie umožňuje reprezentáciu doménového mena takto:

- Postupnosť znakov, ktorá končí nulovým oktetom (koniec reťazca).
- Ukazovateľ na doménové meno.
- Postupnosť znakov, ktorá končí ukazovateľom.

Ak je v položke RDATA doménové meno a je použitá kompresia, tak v položke RLENGTH sa nachádza dĺžka zkomprimovaného názvu, nie celého.

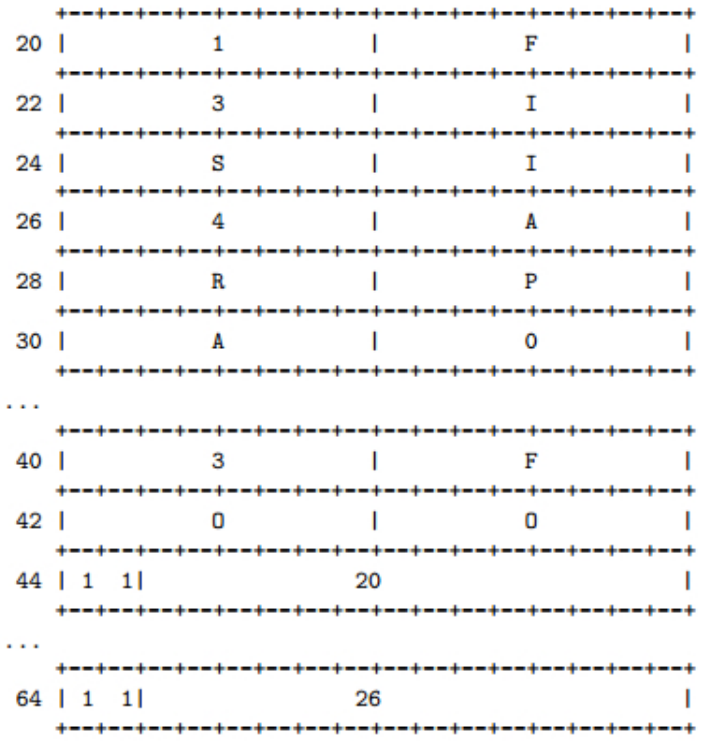
Ako to celé funguje si ukážeme na nasledujúcom príklade. Paket obsahuje doménové mená F.ISI.ARPA, FOO.F.ISI.ARPA, ARPA, a root. Ak budeme ignorovať ostatné položky v pakete, tieto doménové mená môžu byť reprezentované takto 4.2:

Doménové meno pre F.ISI.ARPA začína na offsete 20. Doménové meno FOO.F.ISI.ARPA začína na offsete 40. Táto definícia nám umožňuje využiť ukazovateľ pre spojenie FOO a predchádzajúceho mena F.ISI.ARPA. Doménové meno ARPA je definované na offsete 64 a ukazuje na doménové meno F.ISI.ARPA, konkrétne na octet 20, kde meno ARPA začína na octete 20. Root meno je definované jedným octetom, ktorý je nulový a nachádza sa na pozícií 90, root meno neobsahuje žiadne znaky.

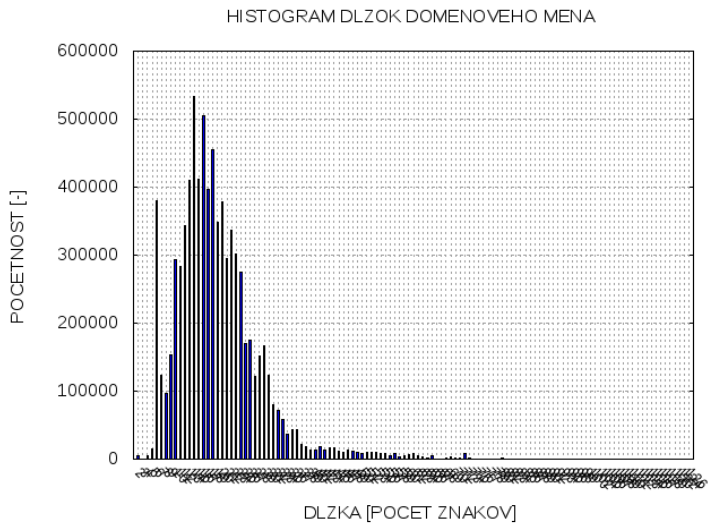
4.3 Spracovanie intervalov pre histogramy

Pri spracovaní veľkého množstva údajov môže nastať problém pri vytváraní histogramu. Napríklad pri spracovaní dĺžok doménového mena môže byť zastúpenie niektorých dĺžok veľmi veľké oproti ostatným, alebo dĺžky doménových mien môžu zaberať veľmi široký rozsah a tým pádom sa stáva histogram veľmi neprehľadný, nečitateľný 4.3.

Tento problém nám pomôže vyriešiť pravidlo, že hodnoty na X-ovej osi rozdelíme do intervalov. Podľa Sturgisovho pravidla 4.3 vieme vypočítať koľko intervalov potrebujeme vytvoriť.



Obrázek 4.2: Príklad kompresie. Zdroj: [12].



Obrázek 4.3: Histogram dĺžky doménových mien bez úpravy.

$$J = 1 + 3.3 * log(n)$$

kde **J** je počet intervalov

n je celkový počet položiek, v našom prípade dĺžok

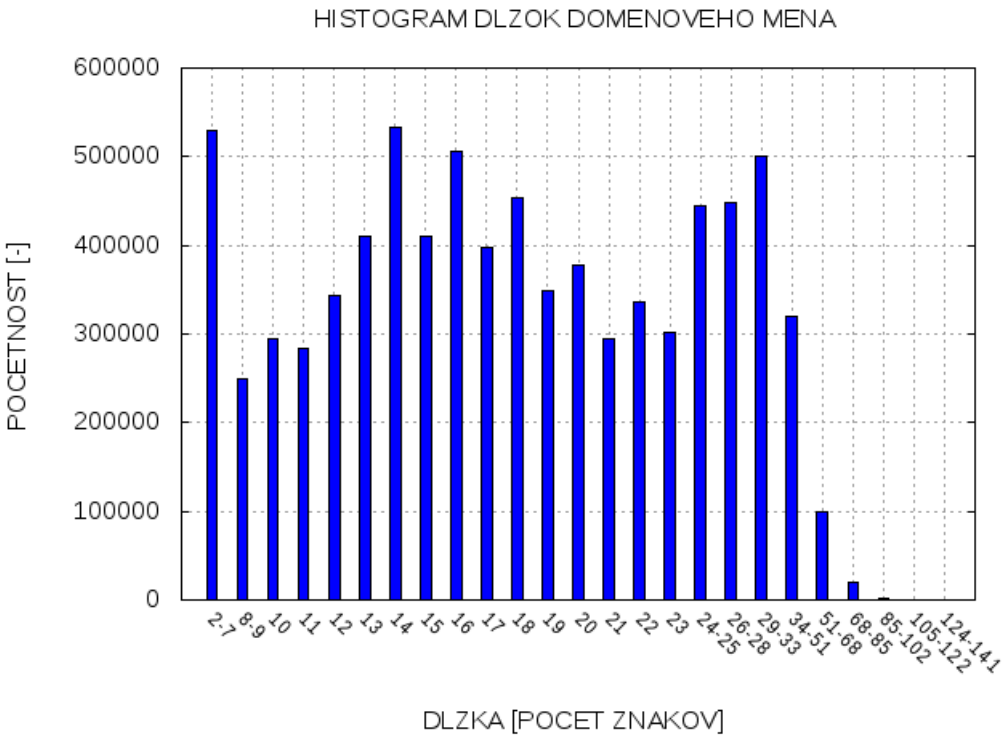
Z tohto výpočtu si vieme ďalej vypočítať, koľko položiek má byť v jednom intervale.

$$P = \frac{n}{J}$$

Ak niektoré z položiek, ktoré nasledujú za sebou na X-ovej osi budú obsahovať veľmi veľké početnosti, môže to spôsobovať to, že niektoré časti grafu budú veľmi malé. Je vhodné aby sa tento stav ošetril.

V histograme sa zistí najväčšia početnosť danej položky a tá bude považovaná za maximálne možnú zobrazenú hodnotu na Y-novej osi. Z tohto vyplýva aj ďalšia výhoda a to, že položky ktoré budú obsahovať najväčšiu početnosť tak sa v histograme zobrazia samostatne, nebudú predstavovať intervaly ale len hodnotu danej položky. Na obrázku 4.4 vidíme rozdiel oproti obrázku 4.3 po použití tejto metódy. Túto metódu budem používať pri kreslení všetkých histogramov v tomto projekte.

V tejto podkapitole som čerpal prevažne z [4].



Obrázek 4.4: Histogram dĺžky doménových mien po úprave.

4.4 Výstup programu

Program vypíše na štandardný výstup výsledky vybraných štatistík. Výsledky sú v textovom formáte. Formát štatistík je rôzny. Na 4.5 sú uvedené príklady. Ak sa z vybraných štatistík dá zostrojiť graf, tak do súboru sa zapíšu hodnoty na jeho vytvorenie. Dáta pre zostrojenie grafov sú vo formáte:

X_0 Y_0

X_1 Y_1

...

X_n Y_n

kde X sú hodnoty na x-ovej osi, Y sú hodnoty na y-novej osi

POMER MEDZI Questions/Answers

Pomer vsetkych dotazov(6813180)/odpovedi(28767828) v prilozenom subore je 0.24

PRIEMERNA VELKOST PAKETOV

Priemerna velkost vsetkych DNS packetov je 3531.93 B

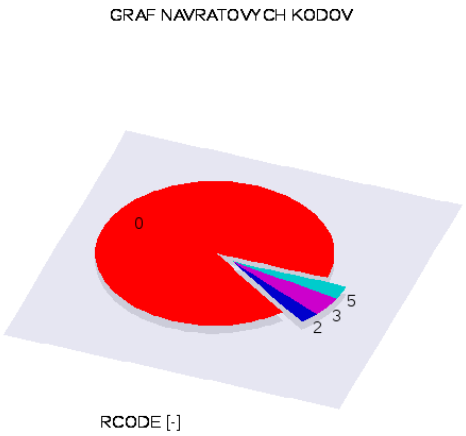
Priemerna velkost QUERY paketov v subore je 3791.36 B

Priemerna velkost RESPONSE paketov v subore je 2689.28 B

Pomer medzi IPv4 (8 478 203) / IPv6 (432 569) adresami je 19.600

Obrázek 4.5: Príklad výstupov

Súbory, ktoré obsahujú dáta pre zostrojenie sa nachádzajú v adresári `./outs`, pomenové sú podľa názvu príslušnej štatistiky (skrátene). Jedná sa o súbory s príponou `.dat`. Ukážka výstupného grafu generovaného skriptom 4.6.



Obrázek 4.6: Ukážka výstupného grafu.

4.5 Skript pre generovanie grafov

Tento skript generuje z výstupných súborov programu grafy. Jedná sa o skript napísaný v programovacom jazyku `Shell`. Skript využíva nástroj na generovanie grafov popísaný vyššie, viď. 2.8.

Pri spustení skript prehľadá adresár `./outs` a ak narazí na súbor, ktorí odpovedá niektorému zo vstupných súborov pre zostrojenie grafu, vytvorí súbor s príslušným grafom. Názov súboru s grafom sa bude volať tak, ako vstupný súbor pre daný graf, plus sa k názvu pridá dátum a čas vytvorenie vo formáte YYYY-MM-DD_shh-mm-ss. Čiže názov výstupného súboru s grafom bude vyzerať `*_YYYY-MM-DD_hh-mm-ss.png`.

Kapitola 5

Testovanie

V tejto časti sa budem zaoberať testovaním implemetnovaného nástroja. Testovať budem časovú a pamäťovú náročnosť programu. Testovanie prebehne na troch súboroch, súbor typu Netflow, CSV a PCAP. Pri meraní časovej náročnosti som využijem nástroj `time` a pre meranie pamäťovej náročnosi programu som zvolil program `valgrind`, s parametrom `--tool=massif --stacks=yes`. Každý zo súborov bude spustený s určitými parametrami, čiže bude sa merať náročnosť aplikácie pri rôzne zvolených štatistikách.

Testy budú vykonávané na PC s týmito parametrami:

`Procesor: Intel Core 2 Duo CPU P8400, 2.26GHz`

`Pamäť: 4 GB`

`Operačný systém: Ubuntu 14.04 LTS 32-bit`

5.1 Test Netflow

Ako bolo popísané vyššie 9, nástroj dokáže z tohto typu súboru vytvoriť len obmedzený počet štatistík z dôvodu obmedzeného množstva položiek. Na testovanie bude použitý súbor `nfcapd_anon.20150222_1`. Program bude spustený s nasledujúcimi parametrami:

- Test 1: `dns_analysis --netflow nfcapd_anon.20150222_1 -r`
- Test 2: `dns_analysis --netflow nfcapd_anon.20150222_1 -b`
- Test 3: `dns_analysis --netflow nfcapd_anon.20150222_1 -r -b -i -p`

	Časová náročnosť [s]	**Pamäťová náročnosť [B]**
Test 1	33.138	11,089,696
Test 2	33.062	11,613,700
Test 3	33.297	11,614,708

Tabulka 5.1: Hodnoty namerané pri jednotlivých testoch.

Z tabuľky 5.1 môžme povedať, že nástroj sa pri vytváraní štatistík správal takmer rovnako. V časovej a pamäťovej náročnosti sú len drobné odchýlky. Je to spôsobené tým, že operácie, ktoré sa vykonávajú pri získavaní údajov pre štatistiky sú podobné.

	Časová náročnosť [s]	Pamäťová náročnosť [B]
Test 1	29.014	13,480
Test 2	129.469	537,776
Test 3	95.803	13,396
Test 4	402.897	537,696
Test 5	499.989	537,772

Tabulka 5.2: Hodnoty namerané pri jednotlivých testoch.

5.2 Test CSV

V tomto príde bude využitý väčší počet štatistik, pretože v súbore sa nachádzaju dáta z aplikačnej vrstvy DNS. Na testovanie bude použitý súbor `dns_ipfix_14042015`, pre jednotlivé testy bude program spustený s týmito parametrami:

- Test 1: `dns_analysis --csv ipfix_14042015 -r`
- Test 2: `dns_analysis --csv ipfix_14042015 -b`
- Test 3: `dns_analysis --csv ipfix_14042015 -r -p -i`
- Test 4: `dns_analysis --csv ipfix_14042015 -h -n -d -o -t -l -b`
- Test 5: `dns_analysis --csv ipfix_14042015 -h -n -d -o -t -l -b -r -p -i`

Z tabuľky 5.2 vidíme, že časová náročnosť spracovania tohto súboru je zavistlá hlavne od počtu vybraných štatistík. Je to spôsobené tým, že pri spracovaní CSV súboru sa pracuje s reťazcami a ich spracovanie je pre procesor pomerne časovo náročná operácia. Časova náročnosť úmerne rastie s počtom vybraných štatistík. Čo sa týka pamäťovej náročnosti programu, tak program v testoch 2, 4 a 5 využíva k uchovaniu dát veľké pole, čo sa odzrkadlilo aj na výsledkoch.

5.3 Test PCAP

Tento test som sa rozhodol otestovať na súbore, kde je zachytený DNS útok. Jedná sa o útok typu DDoS[1]. Tak isto, ako v predchádzajúcih prípadoch bude odmeraná časová a pamäťová náročnosť programu, k tomu ešte pridám výsledky z niektorých štatistík, podľa ktorých by sme mohli povedať, že sa jednalo o DoS útok. Program bude spustený s nasledujúcimi parametrami:

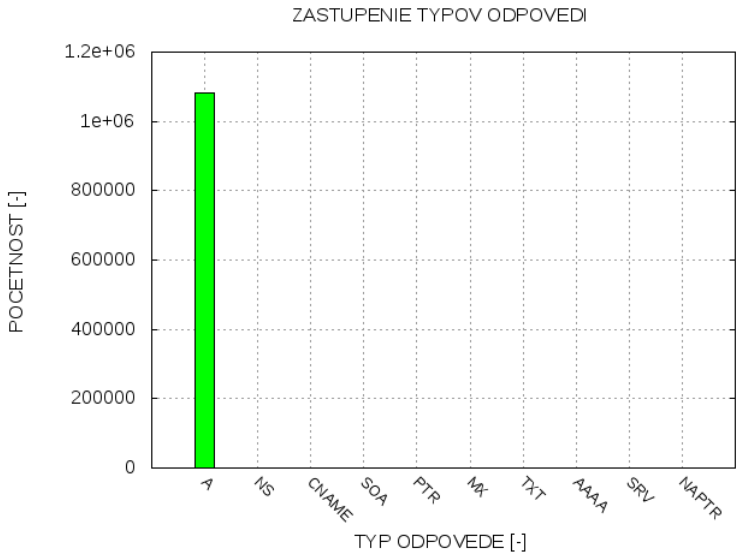
- Test 1: `dns_analysis --pcap attack.pcap -r`
- Test 2: `dns_analysis --pcap attack.pcap -h`
- Test 3: `dns_analysis --pcap attack.pcap -q -r -p`
- Test 4: `dns_analysis --pcap attack.pcap -h -n -d -o -t -l -b`
- Test 5: `dns_analysis --pcap attack.pcap -r -d -o -t -l -b -q -r -p -h`

	Časová náročnosť [s]	**Pamäťová náročnosť [B]**
Test 1	5.632	69,820
Test 2	5.630	71,004
Test 3	5.635	70,996
Test 4	5.698	602,320
Test 5	5.739	602,320

Tabulka 5.3: Hodnoty namerané pri jednotlivých testoch.

V tabuľke 5.3 vidíme, že časová náročnosť programu je vo všetkých testovaných prípadoch takmer rovnaká. Čo o pamäťovej náročnosti sa už povedať nedá, medzi prvými troma testami sú len malé rozdiely. Veľký rozdiel je však oproti testom 4 a 5. Je to spôsobené tým, že na uchovanie dát pre niektoré štatistiky je použité veľké dynamické pole, ktoré sa alokuje na halde.

Ako bolo spomenuté vyššie, jednalo sa o PCAP súbor, ktorý obsahoval DDoS útok. Z grafu 5.1 vidíme, že sa jedná najmä o záznamy typu A, ostatné záznamy sú voči týmto záznamom zanedbateľne. Z toho vyplýva, že bolo zachytených mnoho paketov s odpoveďami na A záznam. Nebolo by to nič nezvyčajné, keby pomer dotazov a odpovedí nebol 0.00024 5.2. To znamená, že jedna strana zasielala veľké množstvo odpovedí, ktoré neboli vyžiadané. Tieto dva ukazovatele nám poslúžili na zachytenie tejto sieťovej anomálie.



Obrázek 5.1: Histogram typov odpovedí v súbore.

```
POMER MEDZI Query/Response PAKETMI
```

Pomer QUERY(355)/RESPONSE(1469768) paketov je 0.00024

Obrázek 5.2: Výpis pomeru medzi Query/Response paketmi.

[1]Distributed Denial of Service

Kapitola 6

Záver

Táto práca sa zaoberá analýzou zachytenej DNS prevádzky. V práci sú popísané základy počítačových sietí, systém DNS a jeho hlavné úlohy, ďalej je vysvetlená dôležitosť monitoringu siete a akú úlohu v tom hrajú sieťové toky.

Pre zvládnutie tejto práce bolo nutné podrobnejšie naštudovať službu DNS, dôležitosť monitoringu siete, možnosti formátov pre zachytávanie tokov Netflow a IPFIX a v poslednom rade aj formát PCAP. Ďalšia časť práce sa venuje analýze vstupných dát, dostupnosťou položiek z jednotlivých formátov pre analýzu a výberom položiek, z ktorých sa budú vytvárať štatistiky.

Ďalej je časť práce venovaná návrhu daného nástroja z pohľadu použiteľnosti, kde cieľom bolo dosiahnuť čo najgeneralizovanejší nástroj, ktorý by bol vo výsledku, čo najviac parametrizovateľný, čo je pri programoch, ktoré spracovávajú veľký objem dát dôležitým faktorom. Pre implemtáciu bolo potrebné naštudovať knižnice na spracovanie Netflow a PCAP v jazyku C/C++.

Posledná kapitola sa venuje testovanie vytvoreného nástroja. V testoch bola sledovaná pamäťová a časová náročnosť implementovaného nástroja. K jednému testu bol pridaný aj výstup z programu, pretože vstupné dáta obsahovali útok na službu DNS. Taktiež boli zhodnotené výsledky jednotlivých testov.

Vytvorená aplikácia dokáže zo zachytenej DNS prevádzky vytvoriť štatistiky, ktoré môžu byť nápomocné pri monitoringu siete, pri zisťovaní vyťaženosti siete, zisťovaní veľkosti prenášaných dát po sieti a aj k zachyteniu sieťových anomálií ako sú napríklad útoky. Aplikácia môže byť dobrým pomocníkom pre správcov siete. Jednoducho a rýchlo sa dokážu dostať k tomu, čo sa na sieti deje a z toho vyvodiť prípadné opatrenia alebo predísť nehomogénnym stavom prevádzky.

Existuje aj priestor pre možné vylepšenia programu. Najžiadanejším vylepšením je podpora spracovania IPv6 paketov v súboroch PCAP. Ďalšie vylepšenie by mohlo byť pridanie ďalších štatistík, napríklad výpis podsietí, z ktorých prebehla komunikácia alebo veľkosť prenesých dát zo všetkých podsietí.

Literatura

[1] Anukool Lakhina, M. C.: Mining Anomalies Using Traffic Feature Distributions. [Online].
URL http://www.ics.forth.gr/mobile/Papers/Mining%20Anomalies%20Using%20Traffic%20Feature%20Distributions.pdf

[2] Bush, R.: Root Name Server Operational Requirements. RFC 2870. Technická zpráva, June 2000.
URL https://tools.ietf.org/html/rfc2870

[3] Břehovský, P.: *Praktický úvod do TCP/IP*. České Budějovice : KOPP, 1997, ISBN 80-85828-29-4, 108 s.

[4] Cimbala, J. M.: Histograms. [Online], 2014.
URL https://www.mne.psu.edu/me345/Lectures/Histograms.pdf

[5] František Jakab, A. P.: Optimalizácia monitorovania sieťovej prevádzky. *Acta Informatica Pragensia*, ročník 2, č. 1, 2013, ISSN 1805-4951.

[6] Garcia, L. M.: Programming with Libpcap - Sniffing the Network From Our Own Application. *Haking*, ročník 3, č. 2, 2008, ISSN 1733-7186.

[7] InetDaemon: DNS Forward Resolution. [Online], Apr 2013.
URL http://www.inetdaemon.com/tutorials/internet/dns/operation/resolution/forward/forward.shtml

[8] Institute, I. S.: TRANSMISSION CONTROL PROTOCOL. Technická zpráva, September 1981.
URL https://www.ietf.org/rfc/rfc793.txt

[9] Kováčik, M.: *Detekce síťových anomálií a bezpečnostních incidentu s využitím DNS dat,pojednání k tématu disertační práce*. Dizertační práce, FIT VUT v Brně, Brno, 2014.

[10] Matoušek, P.: *Síťové aplikace a jejich architektura*. Akademické nakladatelství, VUTIUM, 2014, ISBN 978-80-214-3766-1, 396 s.

[11] Microsoft: Understanding zones and zone transfer. [Online], 2005.
URL https://technet.microsoft.com/en-us/library/cc781340(v=ws.10).aspx

[12] Mockapetris, P.: Domain names - implementation and specification.RFC 1035. Technická zpráva, November 1987.
URL https://www.ietf.org/rfc/rfc1035.txt

[13] Morimoto, R.: Zone Transfers. [Online].
URL http://flylib.com/books/en/4.34.1.117/1/

[14] Nondek, L.: *Internet a jeho komerční využití*. Praha : Grada, 2000, ISBN 80-7169-933-0, 117 s.

[15] Patterson, M.: What is IPFIX vs. NetFlow v9? [Online], 2009.
URL https://www.plixer.com/blog/netflow/what-is-ipfix-vs-netflow-v9/

[16] Pillai, S.: Difference between iterative and recursive dns query. [Online].
URL http://www.slashroot.in/
difference-between-iterative-and-recursive-dns-query

[17] Popescu, D.: Networking fundamentals tutorial - OSI and TCP/IP protocol stacks. [Online], 2013.
URL http://www.ittrainingday.com/2013/01/
networking-fundamentals-tutorial-osi.html

[18] Postel, J.: User Datagram Protocol. Technická zpráva, 28 August 1980.
URL https://www.ietf.org/rfc/rfc768.txt

[19] Roolvink, S.: Detecting attacks involving DNS servers : A netflow data based approach. Technická zpráva, 2008.
URL http://essay.utwente.nl/58497/

[20] SourceForge: NFDUMP. [Online], 2014.
URL http://nfdump.sourceforge.net/

[21] SourceForge: Gnuplot. [Online], 2015.
URL http://gnuplot.sourceforge.net/

[22] Wong, E.: Network Monitoring Fundamentals and Standards. [Online], Aug 1997.
URL http://www.cse.wustl.edu/~jain/cis788-97/ftp/net_monitoring/

Příloha A

Obsah CD

`Data/` - adresár obsahuje data na testovanie

`dns_analysis/` - zdrojové kódy aplikácie, Readme

`latex/` - zdrojové kódy tohto dokumentu

`xhmela00.pdf` - pdf tohto dokumentu

Příloha B

Manual

K fungovanie aplikácie je potrebné mať nainštalovanú knižnicu Libpcap. Získame ju zadaním príkazu `sudo apt-get install libpcap0.8 libpcap0.8-dev libpcap-dev`.

Ďalej je potrebný `nfdump`. V čase tvorby programu vo verzii `nfdump-1.6.13`, dostupný na `http://sourceforge.net/projects/nfdump/files/stable/nfdump-1.6.13/`. Návod na inštaláciu tohto nástroja sa po rozbalení stiahnutého súboru nachádza v súbore INSTALL.

Po preložení príkazom `make`, je program možné spustiť s týmito parametrami:

- `--pcap <názov súboru>` - pre spracovanie PCAP súboru.
- `--netflow <názov súboru>` - pre spracovanie Netflow súboru.
- `--csv <názov súboru>` - pre spracovanie CSV súboru.

  Tieto tri parametre nie je možné kombinovať, je možné spracovať iba jeden súbor naraz.

- `--help` - vypíše help
- `-h` - histogram dĺžok doménového mena + priemerná dĺžka doménoveho mena
- `-n` - pomer návratových kódov + koláčový graf hodnôt
- `-d` - pomer typov dotazov + histogram typov
- `-o` - pomer typov odpovedi + histogram typov
- `-q` - pomer Questions/Answers
- `-r` - pomer Query/Response paketov
- `-p` - priemerná veľkosť paketov
- `-t` - histogram TTL hodnôt
- `-r` - histogram RLENGTH + priemerna velkost RLENGTH

- `-b` - histogram zastupenia portov
- `-i` - pomer medzi IPv4/IPv6
- `-e` - vypíše podrobný výpis z prevedených štatistík

Jednotlivé štatistiky neide vykonávať na všetky typy súborov. Výpis možných štatistík pre jednotlivé súbory:

- `PCAP : -h -n -d -o -q -r -p -t -l -b`
- `CSV : -h -n -d -o -r -p -t -l -b -i`
- `NETFLOW : -r -p -b -i`

Pre vytvorenie grafov je potrebné spustiť skript `script.sh`. Najprv skriptu priradíme práva pre spúšťanie. To dosiahneme príkazom `chmod +x script.sh`. Grafy sa vygenerujú do adresára `outs`.

Pre spustenie programu s testami je potrebné skopírovať adresár `Data` a `dns_analysis` na disk. Pre vytvorenie grafu z jednotlivých testov je nutné po každom teste spustiť skript `./script.sh`. Po preložení je program možné spustiť s týmito parametrami:

TEST NETFLOW

```
Test 1: ./dns_analysis --netflow ../Data/nfcapd_anon.20150222_1 -r

Test 2: ./dns_analysis --netflow ../Data/nfcapd_anon.20150222_1 -b

Test 3: ./dns_analysis --netflow ../Data/nfcapd_anon.20150222_1 -r -b -i
-p
```

TEST CSV

```
Test 1: ./dns_analysis --csv ../Data/ipfix_14042015 -r

Test 2: ./dns_analysis --csv ../Data/ipfix_14042015 -b

Test 3: ./dns_analysis --csv ../Data/ipfix_14042015 -r -p -i

Test 4: ./dns_analysis --csv ../Data/ipfix_14042015 -h -n -d -o -t -l -b

Test 5: ./dns_analysis --csv ../Data/ipfix_14042015 -h -n -d -o -t -l -b
-r -p -i
```

TEST PCAP

```
Test 1: ./dns_analysis --pcap ../Data/attack.pcap -r

Test 2: ./dns_analysis --pcap ../Data/attack.pcap -h

Test 3: ./dns_analysis --pcap ../Data/attack.pcap -q -r -p

Test 4: ./dns_analysis --pcap ../Data/attack.pcap -h -n -d -o -t -l -b

Test 5: ./dns_analysis --pcap ../Data/attack.pcap -r -d -o -t -l -b -q -r
-p -h
```