VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

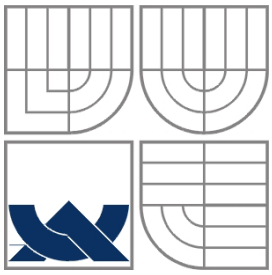
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE ŠKODLIVÝCH DOMÉN POMOCÍ ANALÝZY DNS PROVOZU

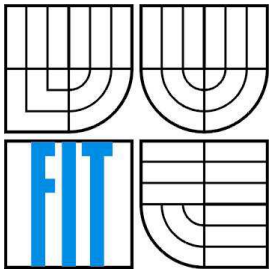
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
VLASTA PODEŠVOVÁ

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

DETEKCE ŠKODLIVÝCH DOMÉN POMOCÍ ANALÝZY DNS PROVOZU

MALICIOUS DOMAIN DETECTION USING ANALYSIS OF DNS TRAFFIC

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
VLASTA PODEŠVOVÁ

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. MICHAL KOVÁČIK

BRNO 2015

Abstrakt

Cílem této bakalářské práce je navrhnout, implementovat a otestovat systém pro detekci škodlivých domén v datech získaných z reálného síťového provozu. Konkrétně se zaměřuje na odhalení činnosti DGA botnetů a to na základě skladby doménového jména. Pro potřeby analýzy skladby doménového jména je součástí řešení systém pro vytvoření modelu ze seznamu legitimních doménových jmen. Implementace této části tak uživateli výsledného systému dovoluje zvolit si vlastní modelová data. Celkově pak tato práce přináší pohled na účinnost implementovaných metod detekce škodlivých domén.

Abstract

The aim of this bachelor's thesis is to design, implement and test a system for malicious domain detection in data sets obtained from real network traffic. It is aimed specifically on detection of DGA botnet activities. This detection is provided by analysis of domain name syntax. Part of the solution is focused at building a model from a set of legal domain names. This model is used for domain name syntax analysis and user of the final system is allowed to choose his own model data. Overall this thesis brings a view on the efficiency of implemented methods of malicious domain detection.

Klíčová slova

DNS, škodlivá doména, botnet, DGA, analýza skladby doménového jména

Keywords

DNS, malicious domain, botnet, DGA, domain name syntax analysis

Citace

Podešvová Vlasta: Detekce škodlivých domén pomocí analýzy DNS provozu, bakalářská práce, Brno, FIT VUT v Brně, 2015

Detekce škodlivých domén pomocí analýzy DNS provozu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Michala Kováčika. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

……
Vlasta Podešvová
20. května 2015

Poděkování

Děkuji vedoucímu mé bakalářské práce panu Ing. Michalu Kováčikovi za poskytnuté rady, které byly cenné pro její zpracování.

Obsah

1 Úvod

V dnešní době nachází Internet využití ve stále více aspektech běžného života lidí a nelze se tomu divit. Ať už se jedná o komunikaci s jinými lidmi, zábavu, nákupy, placení účtů, shánění nejrůznějších informací a další, Internet poskytuje řadu alternativ, z nichž některé představují mnohem schůdnější variantu, než je jejich reálná obdoba. Pro mnohé lidi se tak stal Internet součástí jejich životního komfortu a připadá jim naprosto přirozené přenášet touto sítí mnohdy důvěrné informace. Je ovšem potřeba si uvědomit, že stejně tak jako se neustále rozšiřují možnosti Internetu poskytující běžnému uživateli přínos, mohou se rozšiřovat i možnosti jak prostředí Internetu zneužít k nekalým praktikám. Se stále rostoucím počtem uživatelů Internetu tak rostou požadavky nejen na jeho infrastrukturu, ale i na zajištění bezpečnosti a odhalování potenciálních hrozeb.

Jednou se stěžejních služeb Internetu je systém DNS. Bez jeho fungování by prakticky nebylo možné užívat Internet tak, jak jsme na to zvyklí, a to z toho důvodu, že místo adresování jiného uzlu v síti pomocí doménových jmen by se musely používat IP adresy. Doménová jména jsou totiž ve skutečnosti pouze určitými zástupci IP adres a jejich vzájemný překlad a další služby poskytuje právě systém DNS. Proto je nutné zajistit, aby tento systém nebo jeho část nebyl vyřazen z provozu nějakým útokem. Systém DNS ovšem nemusí být pouze cílem útoku. Díky jeho rozšířenosti mezi všechny uživatele Internetu může být i jeho nástrojem. Systémem DNS se zabývá druhá kapitola.

Jednou z bezpečnostních hrozeb Internetu jsou botnety sloužící ke škodlivým účelům. Takový botnet tvoří ze zařízení infikovaných nějakým malwarem[1] síť, která je pod kontrolou útočníka. Ten ji může zneužívat k celé řadě útoků a nekalých praktik. Některé typy botnetů pak ke svému fungování potřebují systém DNS a vyvinuly si taktiku jak co nejvíce eliminovat možnost jejich vyřazení. Právě na vlastnostech takových botnetů je založena tato práce. Bližší popis je obsahem třetí kapitoly.

Aby bylo možné v síťovém provozu odhalovat útoky jakéhokoli typu, musí být nejprve nějakým způsobem získána konkrétní data. Z tohoto důvodu vznikly prostředky pro monitorování sítí, které poskytují nejrůznější formy takových dat. Výstupem těchto prostředků mohou být např. statistické hodnoty popisující vytíženost síťového zařízení, ale i struktury nesoucí v sobě všechny informace obsažené v zachyceném paketu. Analýzou takto získaných dat tedy lze odhalovat různé síťové anomálie, které mohou značit útok nebo také selhání nějakého prvku sítě. Některé z prostředků pro monitorování sítí jsou přiblíženy ve čtvrté kapitole.

V páté kapitole je navržen systém pro detekci škodlivých domén. Šestá kapitola pak pojednává o jeho implementaci a sedmá o jeho testování.

[1] škodlivý software

2 DNS

Systém DNS[2] je službou aplikační vrstvy síťového modelu TCP/IP a vznikl z jednoduchého důvodu. Číselný formát adresace síťového zařízení v podobě IP adres, který je velmi vhodný pro práci na internetové vrstvě, je zcela nevyhovující na vrstvě aplikační, kdy si běžný uživatel, který s touto vrstvou pracuje, není schopen zapamatovat větší množství IP adres. Proto se na této vrstvě začala v jejích protokolech používat adresace v podobě doménových jmen, které mají textový formát a tudíž jsou pro uživatele přirozeně snáze zapamatovatelné. Jiné služby aplikační vrstvy tak využívají těchto doménových jmen, ale aby se mohly připojit ke vzdálenému síťovému zařízení a jeho službám, stále potřebují znát i IP adresu. A právě překlad doménových jmen na odpovídající IP adresy zajišťuje systém DNS, jehož funkčnost je z tohoto důvodu pro řadu internetových služeb naprosto stěžejní.

Systém DNS poskytuje také další překlady nebo lépe řečeno informace k předané entitě. Jedná se např. o reverzní překlad IP adres na doménová jména, určení poštovního serveru náležejícího dané doméně či určení autorativního serveru dané domény. Bude vysvětleno dále.

2.1 Architektura DNS

Systém DNS spravuje rozsáhlou databázi záznamů DNS, z nichž největší množství připadá na záznamy typu A, tedy překlad doménového jména na IPv4 adresu. Podle údajů společnosti Verisign [2] bylo na konci roku 2014 po celém světě zaregistrováno přibližně 288 milionů doménových jmen. Ke každému doménovému jménu z tohoto počtu musí být někde uložen alespoň jeden záznam typu A, takže je jasné, že tato databáze musí být distribuovaná na více serverů DNS, které jsou k tomuto účelu speciálně vyhrazeny.

Všechna doménová jména jsou uspořádaná do jedné hierarchické stromové struktury a celý tento strom vytváří tzv. prostor doménových jmen. Jeho části jsou v podobě záznamů DNS uloženy na jednotlivých serverech DNS, které pak mezi sebou mají také určitou hierarchii.

Aby klient mohl získat ze systému DNS požadované informace, musí mít na své straně instalovaný tzv. *resolver*, což je program, který formuluje příslušné dotazy na servery DNS. Podle obdržených odpovědí pak rozhoduje, jak postupovat dál.

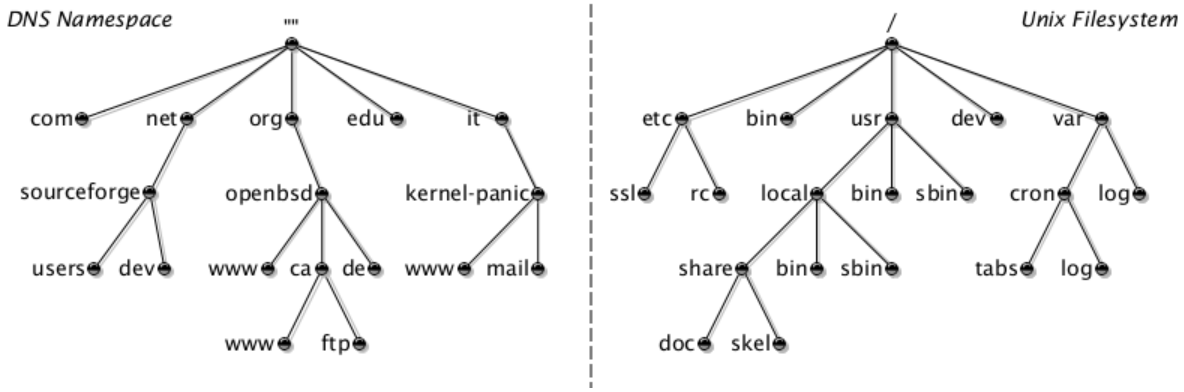
2.1.1 Prostor doménových jmen

Uspořádání doménových jmen do stromu zjednodušuje jejich vyhledávání v celém prostoru, umožňuje je sdružovat podle určité sounáležitosti a logicky rozdělovat jejich správu na jednotlivé servery DNS.

Tento strom má kořen, který se nazývá *root*, který je v zápisu doménového jména reprezentován prázdným řetězcem. Ostatní uzly jsou neprázdnými řetězci, které mají maximálně 63 znaků. Základními povolenými znaky jsou písmena anglické abecedy, číslice a pomlčka, přičemž pomlčka nesmí být na začátku ani na konci řetězce a velikost písmen není rozlišována (DNS je tedy *case-insensitive*). Tento výčet je v některých zemích doplněn o znaky národních abeced (např. azbuky), ale v České republice se rozšíření o znaky české abecedy zatím neplánuje. Doménové jméno je pak cestou od listu ke kořenu stromu a jeho celková délka je omezena na 255 znaků. Jednotlivé názvy uzlů, které tvoří doménové jméno, jsou v jeho zápisu odděleny tečkami. Tento zápis tak teoreticky umožňuje rozdělit doménové jméno až na 127 jednopísmenných uzlů.

[2] Domain Name System

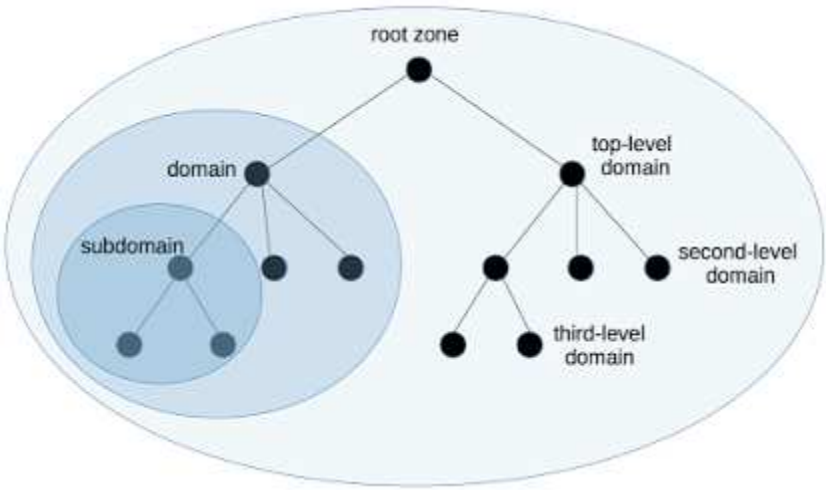
Zápis `www.openbsd.org.`, kde tečka na konci vyjadřuje oddělovač mezi uzly první úrovně stromu a jeho kořenem, se nazývá plně kvalifikované doménové jméno a je určitou analogií absolutní cesty k souboru. Obdobně pokud odebereme jeho část zprava (tedy opačně než je tomu u cesty k souboru), stává se relativním doménovým jménem a jeho interpretace na plně kvalifikované doménové jméno závisí na konkrétním resolveru. Jak už bylo naznačeno, zápis doménového jména se od zápisu cesty k souboru, ale také od zápisu IP adresy, liší tím, že začíná označením konkrétního uzlu sítě a každý další uzel je obecnější než ten předchozí. Tento rozdíl je zobrazen na obrázku 2.1.



Obrázek 2.1: Srovnání stromové struktury prostoru doménových jmen a souborového systému [8]

Doménou nazýváme podstrom prostoru doménových jmen. Mohou obsahovat další subdomény a jsou pojmenované podle doménového jména jejich kořene. Pokud je tento kořen přímým následníkem *root*, nazýváme takovou doménu jako doménu první úrovně, které jsou označovány zkratkou TLD[3]. Mezi tyto domény patří všechny národní domény jako např. cz a také tzv. generické domény jako je např. nejpoužívanější doména první úrovně com. TLD spravuje organizace ICANN[4], která určuje jednotlivé správce těchto domén. Tím je v případě domény cz organizace CZ.NIC.

Díky doménám lze jednotlivé doménové adresy sdružovat do logických celků. Pro www.fit.vutbr.cz je doménou první úrovně doména cz, která sdružuje všechny české domény, druhé úrovně doména vutbr.cz, která sdružuje domény VUT v Brně, a třetí úrovně doména fit.vutbr.cz, která sdružuje zařízení v síti Fakulty informačních technologií. Zároveň však jde také o rozdělení celého stromu na menší části, které mohou být uloženy a spravovány na jednom serveru DNS. Tyto části se nazývají zóny a není nutné, aby jednotlivé subdomény jedné zóny měly společnou doménu.



Obrázek 2.2: Obecné schéma rozdělení prostoru doménových jmen na subdomény [10]

[3] Top Level Domain

[4] Internet Corporation for Assigned Names and Numbers

Speciální doménou je in-addr.arpa, která umožňuje reverzní překlad IP adres na doménová jména. Tohoto překladu se využívá například pro odhalování spamu. Pokud IP adresa, ze které je žádáno o přístup ke službě, nemá příslušný reverzní záznam typu PTR nebo je zpětně získané doménové jméno na seznamu škodlivých domén, přístup bude zamítnut. Uzly tohoto podstromu jsou tvořeny jednotlivými oktety příslušné IPv4 adresy. Jak už ale bylo uvedeno výše, IP adresa je na rozdíl od doménového jména běžně zapisována od kořene stromu k listu. Zápis výsledného doménového jména je tak poněkud netypický, protože IP adresa je zapsána obráceně. Např. 6.96.95.81.in-addr.arpa překládá IP adresu 81.95.96.6. Pro překlad IPv6 adres existuje analogická doména ip6.arpa.

2.1.2 Server DNS

Celý prostor doménových jmen je rozdělen do zón, které jsou v podobě zónových souborů obsahujících záznamy DNS trvale uloženy na tzv. autoritativních serverech DNS. Tyto servery zodpovídají za správu jednotlivých zón a domén v nich obsažených. Dělí se na dva typy:

- **Primární server** je pro každou zónu vždy jen jeden a jen na něm lze provádět úpravy.
- **Sekundární server** obsahuje kopie zónových souborů primárního serveru.

Oba typy mohou poskytovat tzv. autoritativní odpovědi, u kterých by mělo být zajištěno, že odpovídají aktuálnímu stavu. Je tedy potřeba zajistit, aby data sekundárního serveru byla ideálně po celý čas stejná jako na primárním serveru. Každý záznam na sekundárním serveru má proto určenou dobu platnosti, po jejímž uplynutí musí sekundární server vyzvat primární k přenesení případných změn nebo v případě opakovaného selhání aktualizace záznam smazat. Data na sekundárním serveru tak mohou být po určitý čas neaktuální. Aby se tato doba co nejvíce zkrátila, mohou primární servery po provedení změn zaslat svým sekundárním serverům upozornění na tuto skutečnost a sekundární servery pak v odpovědi vyzvou primární, aby zahájili přenos příslušných zónových souborů. Tento přenos probíhá na rozdíl od ostatní komunikace DNS pomocí spolehlivého transportního protokolu TCP.

Existuje ještě jeden typ serverů DNS a to je tzv. záložní server. Takovýto server neuchovává trvale žádné informace. Pokud přijme od klienta dotaz, vyhledá v systému DNS odpověď a tu si pak uloží pro případné další použití. Stejně jako u sekundárních serverů však každá tato odpověď musí mít určenou dobu platnosti, po jejímž uplynutí musí být tyto údaje smazány. Na rozdíl od sekundárních serverů však poskytují záložní servery pouze tzv. neautoritativní odpovědi, které neručí za aktuálnost odpovědi. Většinou jsou tyto servery přidružené k primárním a sekundárním, aby odlehčili jejich zatížení tím, že pro opakované dotazy v určitém časovém úseku není potřeba podstupovat celý proces vyhledávání odpovědi znovu.

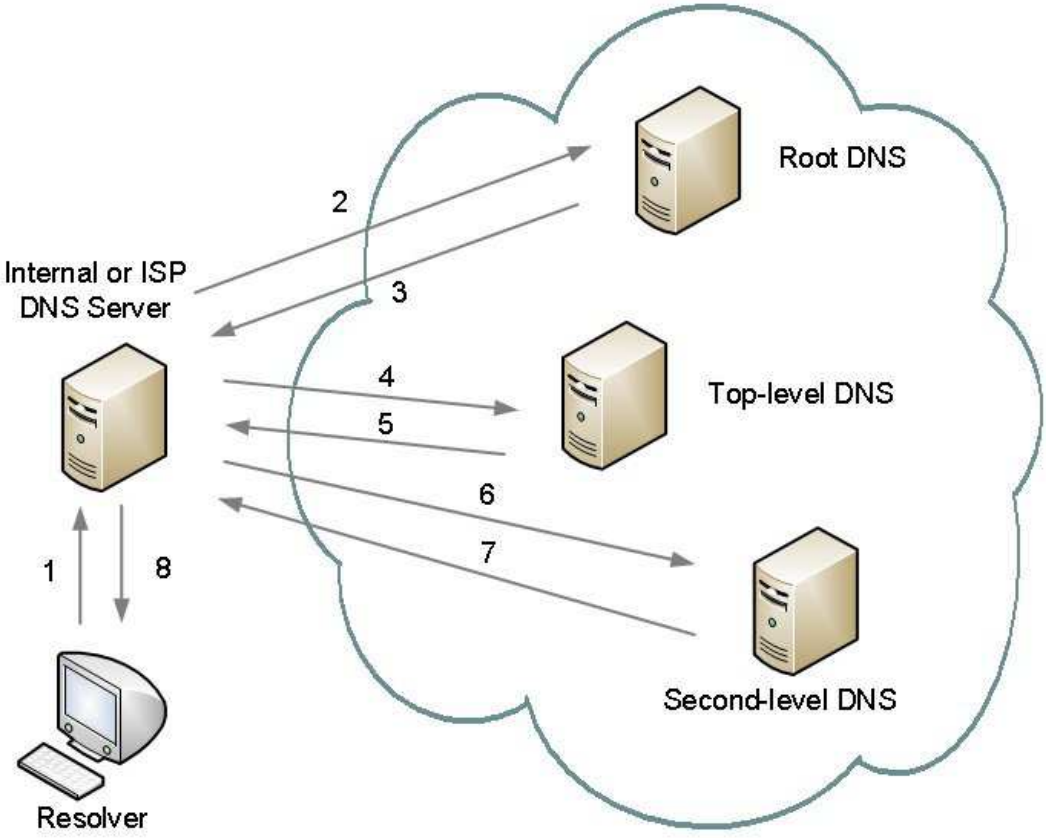
Servery DNS mají také určitou hiearchii a to podle domén, které spravují. Protože servery DNS mohou spravovat více domén, které se v prostoru doménových jmen nacházejí na různých úrovních, je hierarchická úroveň serveru DNS relativní k dané doméně. Na nejvyšší úrovni se nacházejí tzv. kořenové servery, kterých je 13 a na jejich správném fungování závisí celý systém DNS. Tyto servery spravují tzv. kořenové zónové soubory, které obsahují záznamy typu NS určující autoritativní servery pro každou doménu první úrovně. Pro doménu cz jsou to servery [a-d].ns.nic.cz. Primárním serverem je pak a.ns.nic.cz. Podle údajů společnosti CZ.NIC [3] obsluhuje doménu cz a všechny její subdomény 20 552 serverů DNS.

2.1.3 Resolver

Na klientské straně komunikuje se servery DNS resolver. Ten přijímá od aplikací jejich požadavky a transformuje je na odpovídající dotazy, které pak ve formě paketů DNS posílá nejbližšímu serveru DNS. Vždy tak musí být schopen kontaktu minimálně s jedním z těchto serverů. Tato komunikace obvykle probíhá přes nespolehlivý transportní protokol UDP na portu 53. TCP se využívá jen

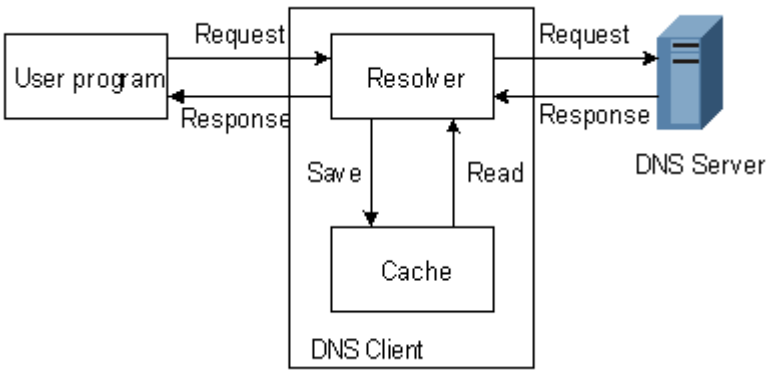
v případě, že přenášený paket DNS je příliš velký a je po cestě fragmentován nějakým síťovým zařízením.

Existují dva způsoby vyhledávání odpovědi v systému DNS. Pokud resolver zašle serveru DNS tzv. rekurzivní dotaz, tak tento server musí odpovědět buď požadovanou informací, nebo chybovou hláškou. Jeden server DNS samozřejmě není schopen obsáhnout odpovědi na všechny možné dotazy, a proto pokud odpověď nezná, musí se dotazovat dalších serverů. Toto dotazování postupuje hierarchicky, takže prvním kontaktovaným je některý kořenový server DNS. K tomuto účelu existuje speciální zóna zvaná *hint*, která obsahuje záznamy právě o kořenových serverech, a každý server DNS musí tento zónový soubor mít, aby bylo zajištěno, že pokud existuje v systému DNS jistá odpověď, bude určitě nalezena. Server DNS tedy zašle kořenovému tentokrát tzv. iterativní dotaz, což znamená, že konktaktovaný server zašle zpět nejlepší možnou odpověď, kterou má k dispozici a sám odpověď na jiných serverech nehledá. V případě kořenového serveru je tak zaslána zpět odpověď obsahující informaci o autoritativním serveru příslušné domény první úrovně. Server DNS tedy pak kontaktuje tento autoritativní server a postup se opakuje, dokud se dotazování nedostane k serveru, který má hledané záznamy. Tento postup je znázorněn na obrázku 2.3.



Obrázek 2.3: Průběh hledání odpovědi na rekurzivní dotaz v systému DNS [11]

Iterativní dotazování samozřejmě může provádět i resolver, čímž se sníží zátěž serverů DNS. Záleží na konkrétním nastavení. Resolver také obvykle má svoji vyrovnávací paměť, do které se ukládají nalezené odpovědi pro případné další použití a pro platnost těchto odpovědí platí to stejné, co pro záznamy uložené na záložních serverech DNS. Práce resolveru je znázorněna na obrázku 2.4.



Obrázek 2.4: Resolver jako prostředník mezi klientským programem a serverem DNS [12]

2.2 Přehled záznamů DNS

Výše již byly zmíněny některé typy záznamů DNS, které tvoří zónové soubory. Zde je uveden přehled několika vybraných typů:

- **SOA** je záznam, který musí být uveden v každé zóně. Obsahuje totiž základní informace o zóně, jako je doménové jméno primárního serveru, e-mailová adresa jejího správce, sériové číslo a časové hodnoty pro obnovu záznamů na sekundárních serverech.
  - Sériové číslo se mění po každé změně zónového souboru na primárním serveru a je důležité pro zajištění správné aktualizace dat na sekundárních serverech. Z důvodu šetření šířky přenosového pásma se totiž často používá tzv. přírůstkového přenosu zón, kdy se nepřenáší celý zónový soubor, ale jen ty části, které jsou jiné než ve verzi zónového souboru, který má aktuálně k dispozici sekundární server. Tato verze je zjistitelná právě z hodnoty sériového čísla.
  - Časová hodnota Refresh určuje dobu, po jejímž uplynutí se sekundární server dotáže primárního na změny daného zónového souboru. Pokud se nelze s primárním serverem spojit nebo selže přenos zóny, sekundární server se o to pokusí znovu po uplynutí hodnoty Retry. Hodnota Expire pak udává maximální dobu, po kterou mohou být záznamy v zónovém souboru platné a hodnota Minimum minimální dobu.
- **NS** je záznam, který určuje jeden nebo více autoritativních serverů pro danou doménu. Alespoň jeden musí být v zónovém souboru uveden. Minimální podoba zónových souborů se tak skládá z jednoho záznamu SOA a jednoho či více záznamů NS. Takovéto zónové soubory se nazývají *stub* a usnadňují vyhledávání v systému DNS.
- **A** je záznam, který uvádí IPv4 adresu příslušející k danému doménovému jménu a je nejčastěji vyhledávaným záznamem.
- **AAAA** je analogií záznamu A pro IPv6 adresování.
- **PTR** je záznam pro revezní překlad IP adres na doménová jména. Tyto záznamy se zapisují do vlastního zónového souboru a doporučuje se mít pro každé doménové jméno příslušný reverzní záznam, aby u některých aplikací nenastaly potíže s autorizací.
- **CNAME** je záznam mapující tzv. aliasy na kanonická jména. K jedné IP adrese může být přiřazeno více doménových jmen. Jen jedno z nich je však kanonickým jménem a toto doménové jméno jako jediné vystupuje v ostatních záznamech DNS. Pokud je tedy vyhledáván v systému DNS některý alias, pak je nejdříve nalezen záznam tohoto typu a až pak je s použitím příslušného kanonického jména vyhledán požadovaný záznam.

- **MX** je záznam, který určuje poštovní servery pro danou doménu. Kromě příslušné doménové adresy tohoto serveru je také uvedena číselnou hodnotou jeho priorita. Poštovních serverů je totiž často více. Pošta je primárně doručována na server, který má nejnižší hodnotu priority, a ostatní servery se používají jako záložní.
- Dalšími používanými záznamy jsou např. NAPTR, TXT, SRV, LOC, DNSKEY, RRSIG, NSEC, NSEC3, DS, a další.

2.3 Zabezpečení DNS

Pro zajištění důvěryhodnosti a zabránění některým útoků na DNS byly vytvořeny bezpečnostní mechanismy, které pracují na základě podepisování pomocí soukromých klíčů. Patří mezi ně TSIG[5] a DNSSEC[6], které jsou v této části popsány.

2.3.1 TSIG

Tato technologie spočívá v podepisování transakcí a s jejím použitím je posílena autentizace uzlů komunikujících mezi sebou v systému DNS, tedy nelze nabourat komunikaci tím, že se útočník vydává za jiný uzel. Uzly, které chtějí spolu komunikovat s tímto zabezpečením, musí mít společný soukromý klíč. Odesílatel tedy ke každému paketu DNS připojí záznam typu TSIG, který obsahuje *hash* podepisovaného paketu, a příjemce pak kontroluje nejenom platnost tohoto *hashe*, ale i název klíče a název hashovací funkce, která byla použita pro zašifrování. Ke každému takovému podpisu je také připojena časová značka, takže útočník nemůže napadnout komunikaci opětovným zasláním odchyceného podepsaného paketu.

Nevýhodou této technologie je, že každá dvojice uzlů musí mít svůj vlastní soukromý klíč. Použití v běžné komunikaci DNS je tedy nemožné, protože většina uzlů by si musela uchovávat obrovské množství takových klíčů. Této technologie se proto využívá zejména pro zajištění bezpečného přenosu zón mezi primárním serverem a jeho sekundárními servery, kdy je počet takto komunikujících uzlů malý. Soukromé klíče mají servery uloženy v konfiguračních souborech.

2.3.2 DNSSEC

Pro zabezpečení běžné komunikace v DNS se používá technologie DNSSEC, která je postavena na podepisování záznamů. V tomto případě se tedy neověřuje uzel, který zprávu zaslal, ale ověřuje se samotný obsah zprávy, tedy přijaté záznamy DNS. Pro tyto účely se využívá dvou klíčů. Jeden je soukromý a zná ho jen autoritativní server příslušný pro danou zónu. Tímto klíčem jsou pro každý záznam v zónovém souboru vytvořeny podpisy v podobě záznamů typu RRSIG. Uzel si pak může ověřit platnost tohoto záznamu pomocí veřejného klíče, který je uložen v záznamu typu DNSKEY. V tomto postupu je tedy využito asymetrické kryptografie, kdy je veřejný klíč vytvořen ze soukromého takovou matematickou funkcí, že je nemožné zpětně získat z veřejné hodnoty vytvořeného klíče hodnotu soukromého klíče jinak než hrubou silou.

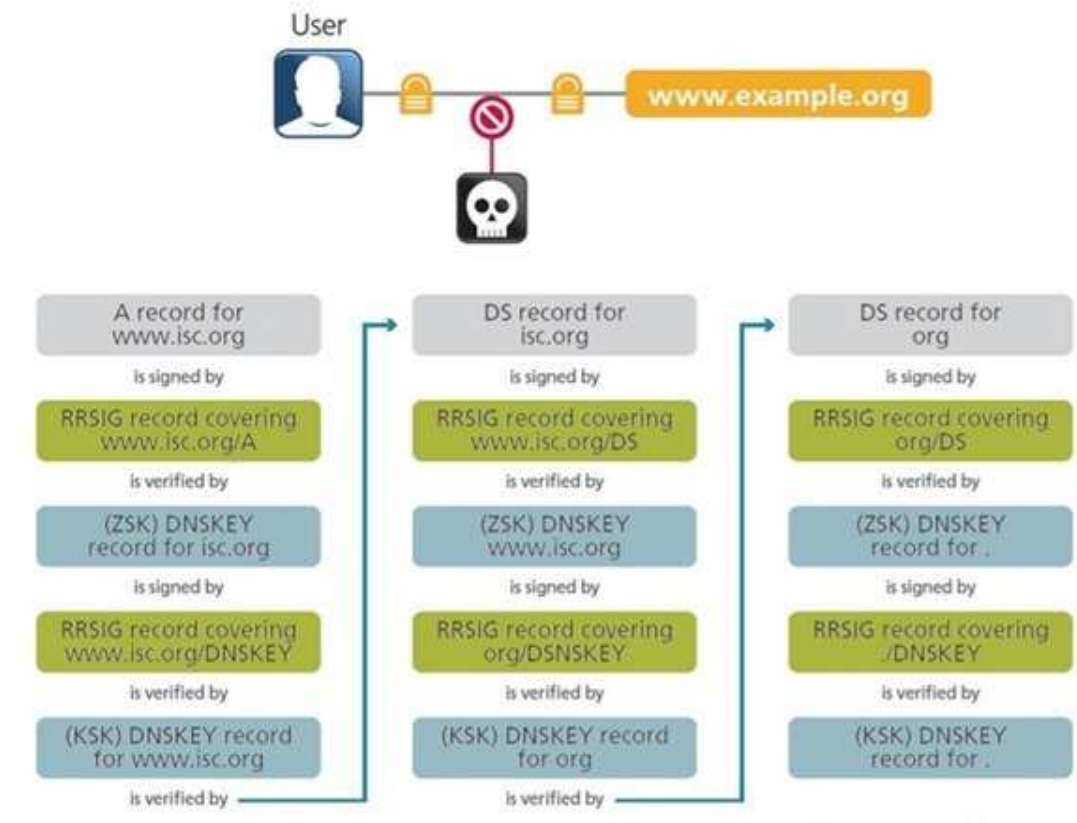
Pokud by ale zabezpečení DNSSEC spoléhalo jen na tuto dvojici klíčů, nebyl by pro útočníka problém vytvořit si vlastní podepsané zónové soubory, jejichž záznamy by se zdály být autentickými, protože totožnost odesílatele se neověřuje. Proto existuje další dvojice soukromého a veřejného klíče,

[5] Transaction SIGnature

[6] Domain Name System Security Extensions

kterou je ověřována platnost dříve zmiňovaného veřejného klíče. Jedna dvojice tedy slouží pro podpis zón a označuje se jako ZSK[7], druhá je určena pro podpis klíče a nazývá se KSK[8].

V každém zónovém souboru jsou tak po dvou obsaženy záznamy typu RRSIG a DNSKEY. První záznam RRSIG obsahuje podpis daného záznamu a jeho platnost lze ověřit pomocí záznamu DNSKEY obsahujícího veřejný ZSK. Druhý záznam RRSIG obsahuje podpis veřejného ZSK a lze jej ověřit pomocí druhého záznamu DNSKEY obsahujícího veřejný KSK. Tento záznam lze pak ověřit pomocí záznamu DS, který je součástí nadřazené zóny a který obsahuje odkaz na tento záznam. Záznam DS je pak podepsán soukromým ZSK nadřazené zóny a takto vzniká tzv. řetězec důvěry, kdy jsou zónové soubory odkazovány podle hierarchie prostoru doménových jmén. Domény, jejichž řetězec důvěry sahá až ke kořenovým zónovým souborům, jsou pak považovány za důvěryhodné. Řetězec důvěry je znázorněn na obrázku 2.5.



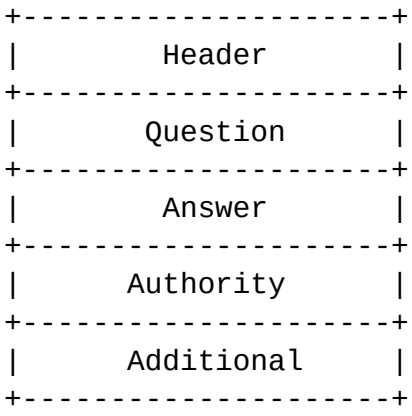
Obrázek 2.5: Řetězec důvěry DNSSEC [4]

Doména cz patří k prvním pěti, které začaly DNSSEC využívat. Stalo se tak v roce 2008. Podle CZ.NIC [3] bylo ke konci roku 2014 zabezpečeno pomocí DNSSEC 452 540 českých domén z celkového počtu 1 173 256, což je více než třetina.

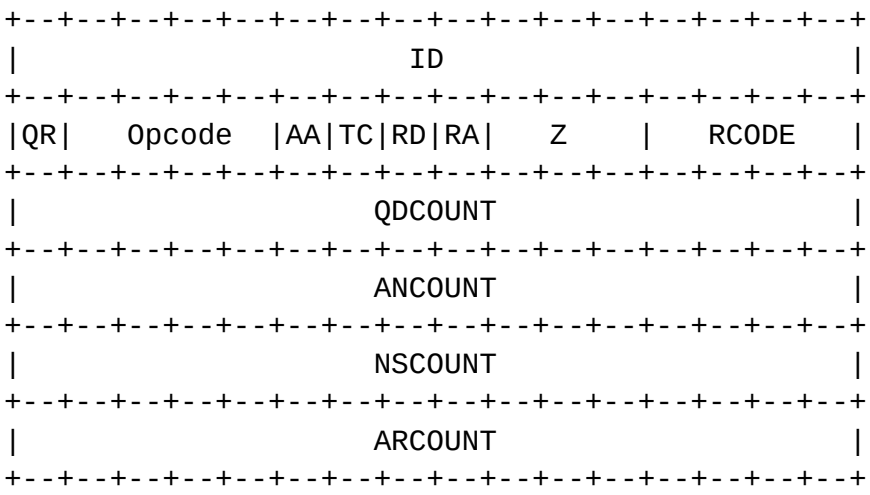
2.4 Paket DNS

Dále v práci se bude s některými položkami paketu DNS pracovat.V této části budou proto tyto položky představeny. Základní struktura paketu DNS je popsána v RFC 1035 a vypadá takto:

[7] Zone Signing Key
[8] Key Signing Key



Hlavička (*Header*) obsahuje následující položky:



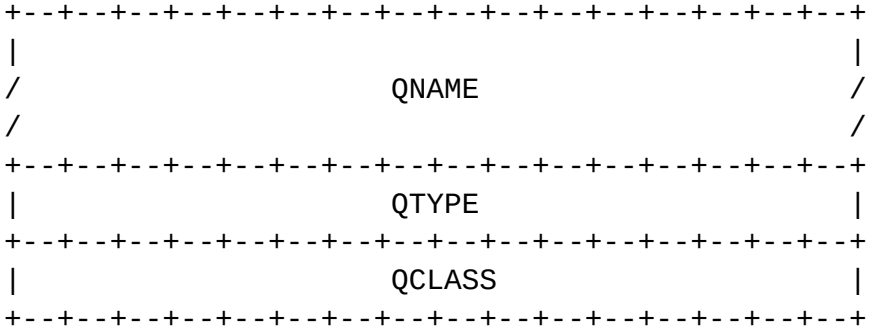
Z těchto položek se dále setkáme s:

- **ID** 16-bitový identifikátor transakce DNS
- **RCODE** návratová hodnota odpovědi DNS vyjádřená číselným kódem:
  - 0 bez chyby
  - 1 špatný formát dotazu
  - 2 chyba na straně serveru DNS
  - 3 dotazovaná doména neexistuje
  - 4 server DNS neimplementuje zpracovaní daného typu dotazu
  - 5 přístup zamítnut

  a další…
- **ANCOUNT** počet vrácených záznamů v odpovědi

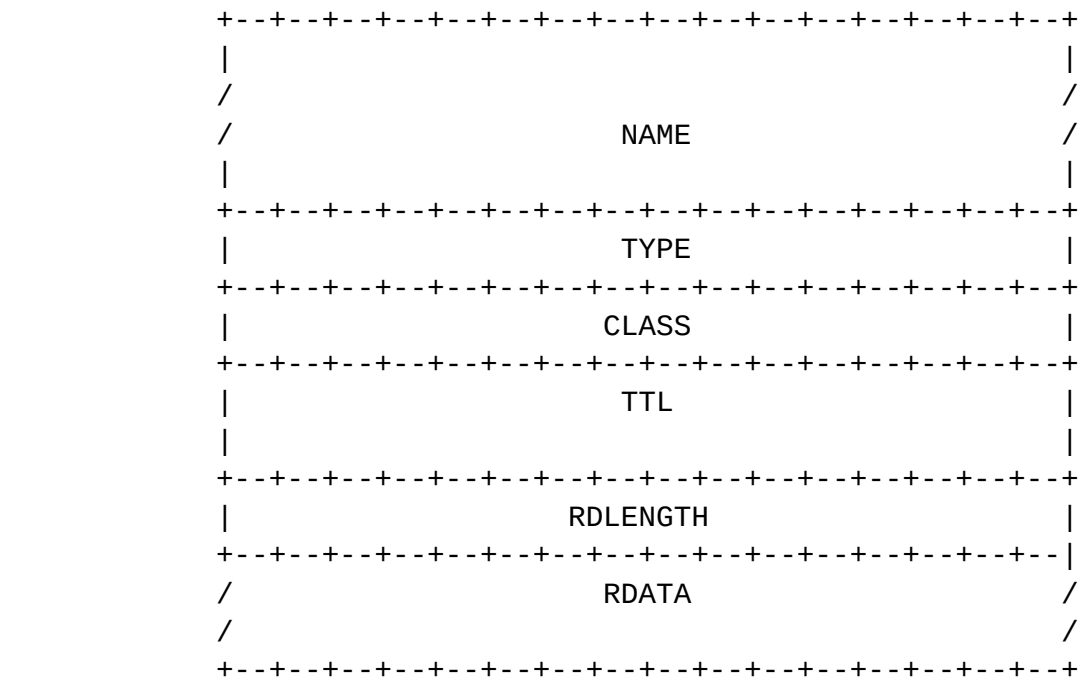
Nutno podotknout, že formát paketu DNS je stejný pro dotazy klienta i odpovědi serveru. Z toho plyne, že položky RCODE a ANCOUNT mají v dotazech vždy hodnotu 0.

Část paketu obsahující dotaz (*Question*) má následující strukturu:



- **QNAME** dotazované doménové jméno.
- **QTYPE** typ dotazovaného záznamu vyjádřený číslenným kódem, např.:
  1 A záznam
  28 AAAA záznam
- **QCLASS** třída dotazovaného záznamu (v prostředí Internetu se používá kód 1)

Část s odpovědí (*Answer*) vypadá takto:



- NAME, TYPE a CLASS jsou stejné jako u části obsahující dotaz.
- **TTL** doba platnosti odpovědi.
- **RDLENGTH** velikost položky RDATA.
- **RDATA** odpověď nebo indikace chyby či dotazu

Část *Authority* obsahuje záznamy NS o autoritativnách serverech a pro tuto práci není důležitá. *Additional* může obsahovat další různé záznamy DNS, které nějak souvisejí s dotazem, ale nejsou to odpovědi. Může být například využita k implementaci EDNS0[9], což je rozšíření protokolu DNS.

2.4.1 EDNS0

Maximální velikost paketu DNS je 512 B. Tato hodnota byla stanovena na začátku vývoje systému DNS a je tak pochopitelné, že se postupem času s navyšující kapacitou přenosových linek a rychlým rozvojem Internetu objevily snahy o zavedení možnosti, aby pakety DNS mohly nést větší objem dat a tudíž i informací. Proto vzniklo rozšíření EDNS0, díky kterému je možno navýšit kapacitu paketu DNS až na 4 kB.

Toto rozšíření dnes používá DNSSEC. Implementováno je pomocí záznamů DNS typu OPT, které jsou svým způsobem záznamy falešnými, protože se nevyskytují v žádných zónových souborech. Tyto záznamy tak lze nalézt pouze v *Additional* části paketu DNS. Hlavička takového záznamu mimo jiné obsahuje bit DO, který je indikátorem toho, zda lze použít DNSSEC nebo ne.

EDNS0 je popsáno v RFC 2671.

[9] Extension mechanisms for DNS

3 Botnet jako bezpečnostní hrozba

Botnet obecně je síť zařízení, na kterých se automaticky spouští určitý typ softwaru, pomocí něhož taková zařízení (nazývaná boti) spolupracují na stejném úkolu. Obecně tedy botnet nemusí sloužit ke škodlivým aktivitám. Často se tak ovšem děje. V takovém případě se boty stávají zařízení infikovaná nějakým malwerem a jsou tak bez vědomí uživatele využívána ke škodlivým aktivitám, tedy např. k rozesílání spamu, DDoS[10] útokům, sbírání citlivých dat uživatele apod. Podle infrastruktury lze rozdělit botnety na dva základní typy, a to na tzv. command&control botnet označovaný jako C&C a tzv. peer-to-peer botnet označovaný jako P2P.

3.1 Command & Control

V rámci takového botnetu jsou boti řízeni serverem, který je přímo spravován útočníkem. Takový server odesílá botům příkazy, případně od nich sbírá získaná data. Fungování C&C serveru je tak stěžejní pro fungování celého botnetu a jeho odhalení je hlavním cílem bezpečnostních analytiků, kteří se snaží botnetům znemožnit jejich činnost. Útočník proto chce docílit toho, aby polohu C&C serveru nebylo možné odhalit.

Pro připojení botů k C&C serveru je využíván systém DNS. C&C server má přidělené doménové jméno a boti musí nějakým způsobem toto doménové jméno získat, aby ho mohli přeložit pomocí serveru DNS na IP adresu. Jednou z možností je, že každý bot má nastavený statický seznam doménových jmen, kde se C&C server může nacházet. Nevýhodou tohoto přístupu pro útočníka ovšem je, že pokud dojde k odhalení celého seznamu, který má bot aktuálně k dispozici, a následnému zablokování v něm obsažených doménových jmen, bot již nemá možnost, jak se k C&C serveru připojit a je tak v podstatě z botnetu vyřazen.

3.1.1 DGA botnet

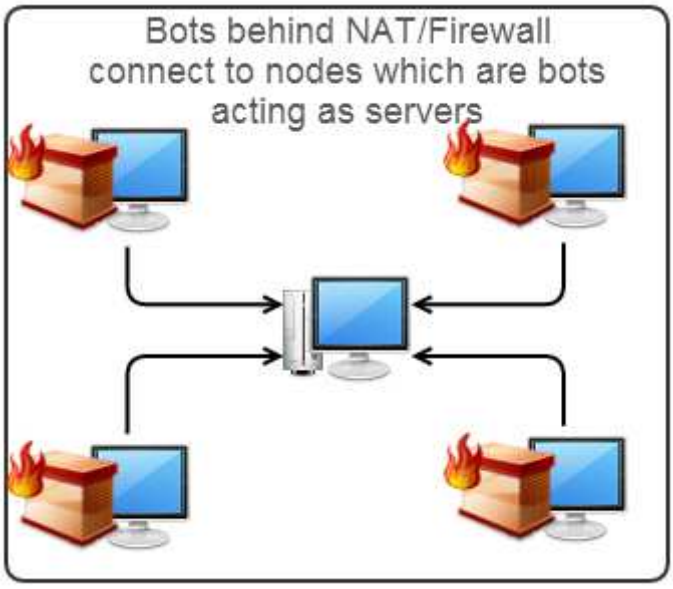
Právě proto vznikly botnety využívající tzv. DGA[11], což jsou algoritmy schopné z daného základu automaticky vygenerovat velké množství doménových jmen. Za pomoci takových algoritmů jsou boti schopni periodicky vytvářet z různých základů (např. aktuálního data a času) nové seznamy doménových jmen. Pokud tedy dojde k tomu, že bot bude mít k dispozici jeden den seznam doménových jmen, z nichž všechny jsou zablokované, neznamená to, že už není schopen připojení, protože další den je vygenerován seznam nový. Tato technika také činí obtížnějším samotné odhalení polohy C&C serveru, protože jeho doménové jméno se pravidelně mění.

Prvním známým malwarem, který tuto techniku využíval a také proslavil, byl Conficker. Na jeho příkladu je také vidět potenciál takových botnetů. Zatímco jeho první verze generovala každý den 250 doménových jmen s pěti možnými TLD, další verze už byly schopné vygenerovat jich 50 000 v rozsahu více jak 110 TLD. Proto také odhalení algoritmu, který stojí za generováním doménových jmen dané rodiny malwaru, nezaručuje nutně to, že takovému botnetu bude zcela znemožněno jeho fungování. Odhalená doménová jména by totiž museli zablokovat všichni registrátoři postižených TLD a je nutno podotknout, že takové pokusy jsou spíše výjimkou. Přesto je odhalení algoritmu pro útočníka nežádoucí, protože v takovém případě se může kdokoliv jiný, tedy i bezpečnostní analytik, pokusit ustavit nový C&C server a převzít kontrolu nad boty, kteří se na něj připojí.

[10] Distributed Denial of Service
[11] Domain Generation Algorithm

3.2 Peer-to-peer

Za vznikem tohoto typu škodlivých botnetů stojí snaha o vytvoření decentralizované sítě nezávislé na jednom bodě, jehož odhalení může vést k zablokování celého botnetu nebo alespoň jeho části. Toho je v případě P2P botnetů dosáhnuto tím, že jako server poskytující příkazy může vystupovat kterýkoliv bot, který může akceptovat příchozí spojení. Ostatní boti, kteří se nacházejí například za překladem NAT nebo firewallem, se pak chovají jako klienti. Uvedené chování je znázorněno na obrázku 3.1.



Obrázek 3.1: Infrastruktura P2P botnetu [6]

Každý bot má k dispozici seznam IP adres některých ostatních botů. Tento seznam získal při svém prvním připojení k botnetu od tzv. *bootstrap* serveru jako výběr z rozsáhlejšího seznamu, který *bootstrap* server spravuje. Tento server je zároveň jediným pevným bodem P2P botnetu. Jeho zablokováním ovšem lze docílit jen toho, že se k botnetu nemohou připojit noví boti, dokud útočník nezprovozní nový *bootstrap* server. Fungování existující struktury botnetu není tímto krokem ohroženo.

Aby útočník mohl botům předat svoje příkazy, stačí mu jen znát IP adresu některého bota vystupujícího jako server a zaslat tyto příkazy jemu. Ten je pak rozešle ostatním botům, jejichž IP adresy zná, a ti udělají pokud možno to samé, a takto se příkazy postupně šíří do dalších částí botnetu. Příkazy útočník zároveň podepisuje svým soukromým klíčem, čímž je někomu, kdo nemá tento klíč k dispozici, znemožněno přebrat kontrolu nad botnetem. A to není jedinou komplikací při pokusech o zneškodnění takového botnetu. Použití této techniky totiž zapřičiňuje to, že je řízení botnetu distribuováno mezi obyčejné uživatelské stanice, a ty nelze všechny najednou zablokovat tak jako odhalené C&C servery.

Existují rodiny malwaru, které používají oba uvedené typy komunikace v botnetu. Jedná se například o některé verze malwaru Zeus, který primárně používá P2P síť, ale pokud jsou všechny pokusy o připojení k jinému botovi neúspěšné, pokusí se o spojení s C&C serverem pomocí DGA.

3.3 Odhalování DGA

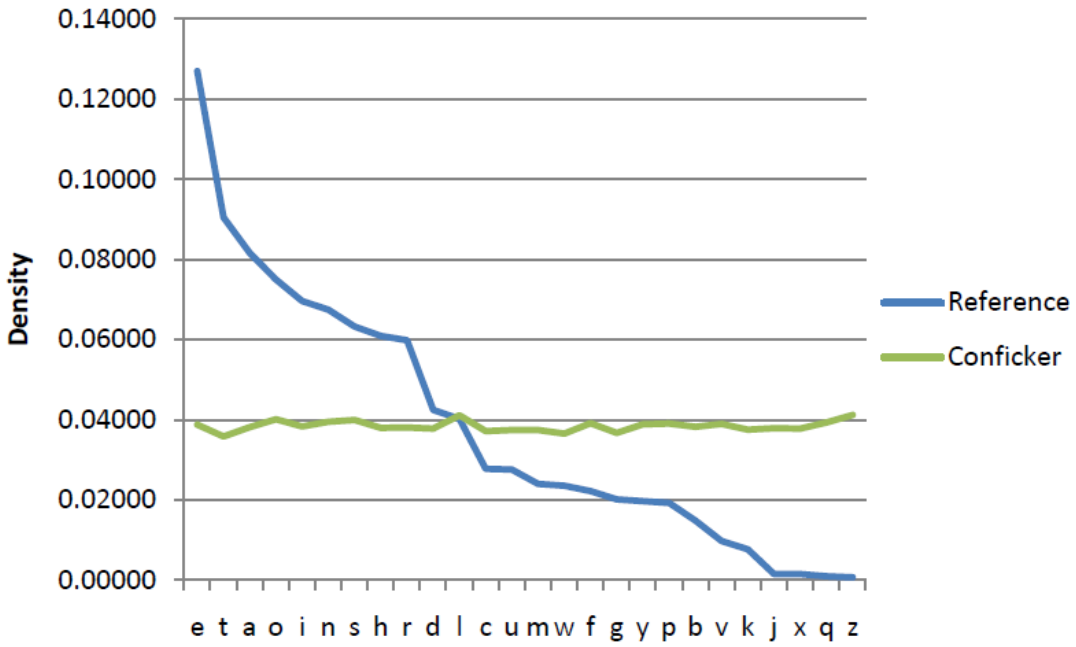
Kvůli uvedeným vlastnostem DGA botnetů jsou tyto botnety mezi útočníky stále velmi oblíbené a jejich činnost představuje značné bezpečnostní riziko v prostředí Internetu. Tato hrozba samozřejmě nezůstává nepovšimnuta a již mnoho studií a odborných článků se věnovalo této problematice. V těchto pracích se objevují v různých obměnách dvě základní techniky detekce působení DGA algoritmu, a to:

- detekce na základě analýzy skladby doménového jména
- detekce na základě výskytu NXDOMAIN[12]

3.3.1 Analýza skladby doménového jména

```
kuwzclqpw.com
hspch.net
sumkuezgsq.info
ibcct.net
pzncbbmy.biz
vjvjdy.net
qicpzx.info
mwekxhj.net
umyar.org
wfhbvr.info
```

Toto je příklad doménových jmen vygenerovaných v botnetu Conficker. Na první pohled je jasné, že tato doménová jména se značně liší od těch, která používá běžný uživatel a toto pozorování tak otvírá cestu k návrhu celé řady testů založených na analýze skladby doménového jména. Autor studie [8] například nabídl srovnání frekvenční analýzy vzorku Confickeru se vzorkem legitimních domén. Výsledek je vidět na obrázku 3.2. Rozložení četnosti znaků je u Confickeru takřka rovnoměrné.



Obrázek 3.2: Frekvenční analýza Confickeru oproti referenčnímu vzorku legitimních domén

[12] Non-existent Domain

Na základě tohoto pozorování pak autor otestoval metodu detekce pomocí počítání váhy frekvenční analýzy slova a došel k výsledkům, které byly nad jeho očekáváním vzhledem k tomu, o jak jednoduchý test jde. Nicméně uzavřel toto téma s tím, že rozhodně nelze očekávat, že všechny DGA produkují podobně rovnoměrné rozložení znaků jako Conficker.

Podobně uvažovali také autoři studie [13], kteří zmínili botnet Kraken a botnet Kwyjibo. DGA těchto botnetů už jsou propracovanější. Kraken klade důraz na použití samohlásek v takovém počtu, který je blízký skutečným slovům, a jejich rovnoměrným rozložením mezi souhlásky se tak snaží o tvorbu vyslovitelných řetězců. K těm ještě navíc připojuje obvyklé přípony anglické abecedy jako –able, -ment, -ly. Kwyjibo se zase naučí ze vzorku skutečných slov sled používání slabik a skládá doménová jména pomocí tohoto naučeného modelu. Obě tyto techniky tak produkují frekvenční analýzy, v nichž už je patrné, že některé znaky jsou protěžovány více. Do frekvenční analýzy skutečného jazyka však mají stále daleko.

Výsledkem této úvahy tedy bylo, že i přesto, že autoři implementovali podstatně složitější testy, než je počítání váhy slova, navrhli také detekci škodlivých domén na základě rozložení četnosti bigramů.

3.3.2 Analýza výskytu NXDOMAIN

Tato detekce byla představena v práci [14] a její podstatou je sledování chování botů využívajících DGA. To se totiž často vyznačuje v provozu tím, že produkuje nadměrný počet odpovědí DNS s návratovým kódem 3, který je příznakem neexistující domény. Právě tento kód je označován jako NXDOMAIN.

Příčina takového chování se dá dobře vysvětlit na příkladě dříve již zmíněné verze Confickeru, která je schopna každý den produkovat 50 000 doménových jmen. Jednotliví boti z tohoto celkového počtu jich produkují pouze 500, takže pokud je C&C server jenom jeden, pak je pravděpodobnost jejich připojení v daný den pouhé 1 %. Z toho plyne vysoká pravděpodobnost, že takový bot zašle 500 dotazů DNS a na všechny se mu vrátí odpovědi značící NXDOMAIN. Výskyt takového jevu v reálném provozu je tak nanejvýš podezřelý.

4 Monitorování síťového provozu

Objem síťového provozu je v dnešní době obrovský a stále narůstá. Z tohoto důvodu není možné síť efektivně spravovat bez použití automatizovaných nástrojů a technik speciálně určených pro sledování sítě a poskytování informací o jejím stavu. Za pomoci těchto nástrojů je správce sítě schopen odhalovat kritická místa sítě a také případné síťové útoky a na základě toho provést příslušná protiopatření.

Tyto nástroje mohou provádět monitorování pasivní nebo aktivní. V případě pasivního monitorování se pouze sbírají data o probíhajícím síťovém provozu a anomálie jsou tak detekovány na základě následné analýzy takto získaných dat nebo na základě chybových hlášení různých síťových aplikací, které mohou být sbírány např. protokolem Syslog. Pokud se tak stane některý prvek sítě nedostupným, lze tuto skutečnost zjistit jen nepřímo, např. ze statistiky ukazující navýšení zátěže jiného prvku. U aktivního monitorování naopak dochází k opakovanému dotazování dostupnosti prvku. Takový nástroj je pak přímým účastníkem provozu.

Nástroje pro monitorování síťového provozu se liší také mírou poskytovaných informací. Velmi rozšířená implementace protokolu SNMP[13] je schopná poskytnout nejrůznější statistiky, s jejichž pomocí lze např. lépe rozvrhnout zátěž jednotlivých prvků sítě. Pro analýzu bezpečnostních incidentů je však nevhodná. Pomocí tohoto protokolu je možné např. sledovat průběh navýšení objemu přenášených dat při DDoS útoku, ale konkrétní informace o tom, co bylo jeho cílem nebo jaký protokol byl k němu zneužit, poskytnout nedokáže. Takovéto informace lze získat jen, pokud jsou přímo sledovány i některé položky přenášených paketů.

4.1 NetFlow

NetFlow je velmi rozšířený nástroj od společnosti Cisco. Tento nástroj umožňuje monitorování tzv. síťových toků. Síťový tok tvoří všechny pakety přenesené v daném časovém úseku, které mají stejnou pětici těchto parametrů:

- zdrojová IP adresa
- cílová IP adresa
- zdrojový port
- cílový port
- protokol

Aby tedy NetFlow mohl pakety sdružovat do toků, musí už přistupovat k jejich konkrétním položkám. Obecně však NetFlow neposkytuje jejich hloubkovou analýzu, ale zpracovává souhrnné informace o toku jako je např. celková velikost přenesených dat apod. To jej sice činí v porovnání s ostatními nástroji rychlým, ale pro navrhovanou detekci škodlivých domén je potřeba získávat z paketů DNS více informací, než NetFlow umožňuje.

4.2 Souborový formát pcap

Do tohoto formátu lze uložit celý paket. Existuje řada nástrojů, které s tímto formátem pracují a jsou tak schopny poskytovat kýženou hloubkovou analýzu paketů. Patří mezi ně např. WireShark nebo

[13] Simple Network Management Protocol

tcpdump. V tomto případě ovšem vystává opačný problém než u NetFlow a to ten, že zpracování celých paketů je pomalé. Ideálním řešením by tak byl nástroj kombinující obě dvě uvedené techniky.

4.3 IPFIX

Přesně takovým nástrojem je IPFIX[14]. Ten vznikl z NetFlow v9 a na rozdíl od něj je standardem IETF[15]. V NetFlow v9 byly představeny šablony, které umožnily vytváření vlastních položek přenášejících údaje o sledovaném paketu. Tato flexibilita je pak základním stavebním kamenem IPFIX a díky ní je umožněno sledování pouze vybraných položek paketu a navýšení rychlosti zpracování oproti nástrojům pracujících s pcap.

[14] Internet Protocol Flow Imformation Export
[15] Internet Engineering Task Force

5 Návrh systému

Základním zdrojem dat pro analýzu škodlivých domén je provoz DNS. Ten je pro účely této práce zachytáván a filtrován od ostatního síťového provozu pomocí nástroje [17], který pracuje s protokolem IPFIX. Tento nástroj zároveň také získaná data anonymizuje, tedy nahrazuje skutečné IP adresy jinými, aby nebylo možné tato data zneužít.

5.1 Vstup

Výsledný soubor tohoto procesu je ve formátu CSV[16], což je jednoduchý textový formát určený k popisu dat. V tomto formátu každý řádek souboru odpovídá jedné položce, která je popsána atributy. Ty jsou od sebe odděleny určeným znakem, který funguje jako oddělovač v celém souboru. V základní podobě je to čárka, z čehož také vychází název tohoto formátu. Mohou být ale použity i jiné oddělovače, např. středník nebo tabulátor. Tento formát je tak v podstatě jednoduchým vyjádřením tabulky.

V konkrétním případě vstupního CSV souboru tohoto systému je oddělovačem čárka a každý řádek odpovídá jednomu zachycenému paketu DNS. Ten je popsán atributy, které odpovídají jeho položkám popsaným v kapitole 2.4. Zde je výčet těchto atributů a jejich protějšků v paketu DNS:

- DNS_ID ID
- DNS_RCODE RCODE
- DNS_ANSWERS ANCOUNT
- DNS_NAME QNAME
- DNS_QTYPE QTYPE
- DNS_CLASS CLASS
- DNS_RR_TTL TTL
- DNS_RLENGTH RDLENGTH
- DNS_RDATA RDATA
- DNS_DO DO

Další atributy jsou vyňaty z ostatních hlaviček paketu (IP, UDP). Z nich jsou nejdůležitější:

- DST_IP cílová IP adresa
- SRC_IP zdrojová IP adresa
- DST_PORT cílový port
- SRC_PORT zdrojový port
- PROTOCOL transportní protokol
- BYTES celková velikost paketu

Analýzou těchto atributů lze detekovat různé anomálie v provozu DNS. Následující tabulka zobrazuje, které atributy se analyzují v navrženém systému a jaké jsou na ně požadavky:

Atribut	
DNS_NAME	vyžadován
DNS_QTYPE	volitelný
DNS_RCODE	volitelný

[16] Comma-separated Values

5.2 Model a jeho volba

Skladba doménového jména může poskytnout celou řadu informací, podle kterých lze rozhodovat, jestli doménové jméno patří škodlivé doméně nebo ne. Aby ale tohle rozhodování mohlo být prováděno, je potřeba mít k dispozici model reprezentující parametry doménových jmen vyskytujících se v legitimním provozu a zkoumat podle něj odchylky daného doménového jména. Otázkou tedy je, jak tento model získat.

5.2.1 Přirozený jazyk

Jak už bylo uvedeno, doménová jména vznikla proto, aby byla snadno zapamatovatelná, a je tak zájmem každého, kdo se chce na Internetu úspěšně prezentovat, aby si skutečně takové doménové jméno zvolil a zaregistroval. Proto lze předpokládat, že budou značně převažovat doménová jména svými parametry blízká přirozenému jazyku. Nabízí se tedy možnost použít jako model některý jazyk. Který jazyk by to ovšem měl být? V prostředí Internetu převládá angličtina, ale díky národním TLD je teoreticky možné zvolit si za model jazyk dané národní domény. A nemusí se to týkat jen národních domén. S příchodem velkého počtu nových generických domén se i u některých z nich může stát, že převládne jiný jazyk než angličtina. Je však nutno zvážit reálné možnosti takového rozdělení. V celém svém měřítku je tato možnost pravděpodobně nerealizovatelná.

Dalším faktem, který je třeba brát v potaz, je to, že takový model nemůže zcela pokrýt veškeré parametry doménových jmen a to už jen z důvodu, že slova žádného přirozeného jazyka neobsahují číslice. Otázkou také je, nakolik je podobnost doménových jmen některé domény k danému jazyku ovlivněna např. tím, že nadnárodní společnosti si registrují své jméno s různými TLD nebo naopak, že si někdo zaregistruje doménové jméno odpovídající jinému jazyku než angličtině pod některou generickou doménou, atd. Drtivá většina jazyků ma vysokou frekvenci samohlásek, což lze tak předpokládat i u doménových jmen, ale skutečně se projeví mezi doménami cz a com rozdíl např. v protěžovaných souhláskách daných jazyků (‘v’ a ‘w’ mezi češtinou a angličtinou)?

5.2.2 Obsah celé domény

Na takové otázky by mohly poskytnout odpovědi modely vytvořené přímo ze všech doménových jmen dané domény. Takové modely už také obsahují data potřebná pro zkoumání všech parametrů doménového jména. I zde je ovšem možnost určitého posunu modelu oproti reálnému provozu a to z toho důvodu, že v doméně může být zaregistrován jistý počet nevyužívaných doménových jmen, které se svými parametry mohou značně vymykat normálu. Navíc možnosti takového systému jsou značně svázány faktem, že pro mnoho domén neexistuje veřejný seznam všech zaregistrovaných doménových jmen.

5.2.3 Alexa Top 1 000 000 Websites

Poslední zvažovanou možností je seznam 1 000 000 nejnavštěvovanějších domén ve světě od společnosti Alexa. U takového seznamu je tedy určitě zaručeno, že všechny v něm obsažená doménová jména jsou používaná a to hojně. Jeví se tak jako nejvhodnější volba, ačkoli existuje možnost, že by nemusel být příliš efektivní pro TLD, které v něm mají malé nebo žádné zastoupení.

5.2.4 Co zvolit?

Během implementace programu jsem používala nejčastěji seznam od Alexy. Rozhodla jsem se ovšem nechat tuto otázku otevřenou a součástí systému je tak i část pro automatické vytvoření modelu

z předaného seznamu doménových jmen. Tím je tedy vyřazena možnost použití přirozeného jazyka jako modelu, což je ale pochopitelné z důvodu uvedené neschopnosti takového modelu obsáhnout všechny znaky používané v doménových jménech.

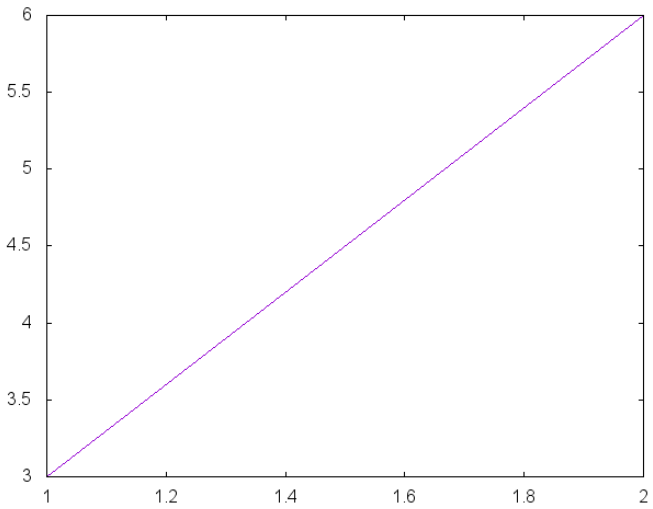
5.3 Utilitita gnuplot

Tato utilita je určená pro tvorbu dvourozměrných a třírozměrných grafů z předaných dat. Soubory určené pro zpracování touto utilitou musí mít následující formát:

```
# X   Y
1.0 3.0
2.0 6.0
```

Řádek začínající znakem '#' označuje komentář. Řádky, které nejsou uvozené tímto znakem, pak nesou konkrétní hodnoty určené ke zpracování. Jednotlivé hodnoty jsou od sebe odděleny některým bílým znakem. Následujícím příkazem se vykreslí z uvedených dat jednoduchý graf odpovídající funkci `y = 3x`:

```
plot "filename" using 1:2 with lines
```



Obrázek 5.1: Graf `y = 3x` vytvořený pomocí utility gnuplot

Výsledné grafy lze ukládat do různých typů souborů nebo zobrazovat přímo v terminálu této utility.

V navrženém systému je tato utilita použitá k automatickému vytvoření grafů o některých částech vytvořeného modelu, jako např. grafu frekvenční analýzy.

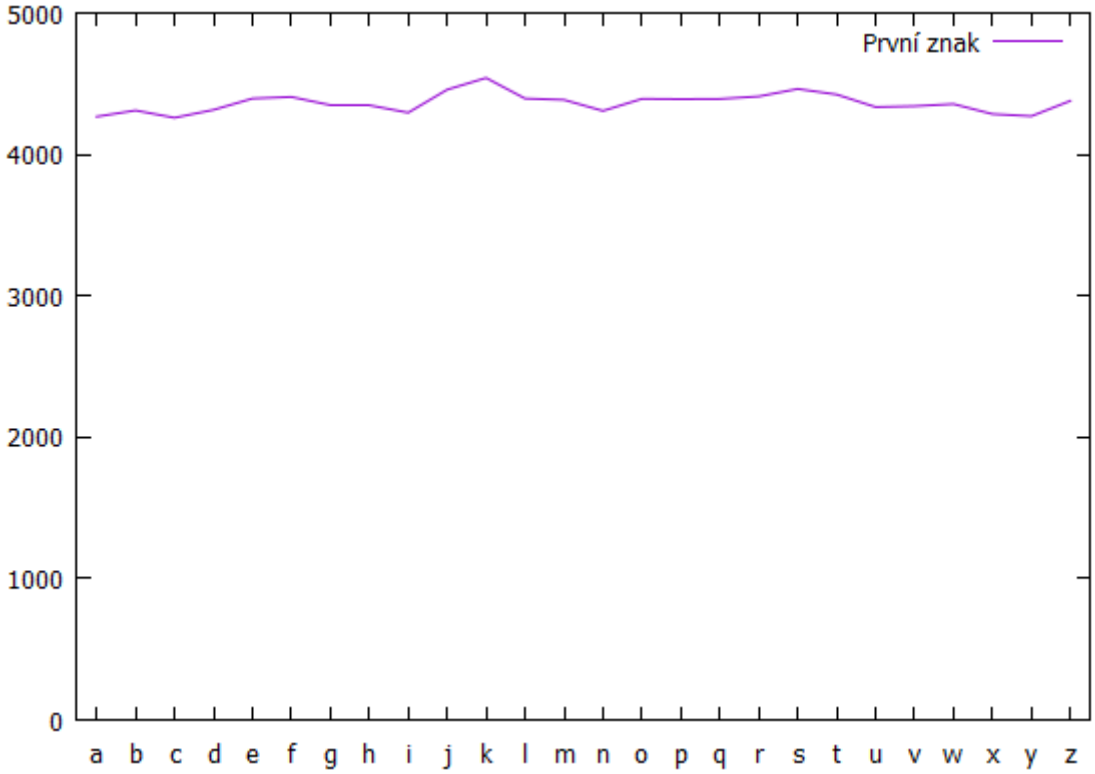
5.4 Detektor škodlivých domén

Navržený detektor škodlivých domén pracuje na základě analýzy skladby doménového jména. Podmínkou spuštění této analýzy je přítomnost atributu DNS_NAME ve vstupním souboru. Atributy DNS_QTYPE a DNS_RCODE jsou nepovinné a slouží např. ke zkoumání toho, jak se mění procento domén detekovaných jako škodlivých v závislosti na výskytu NXDOMAIN apod. Pro tuto analýzu jsou navrženy následující kontroly:

- **Kontrola prvního a posledního znaku**

Myšlenka této kontroly je založena na obrázku 3.2. Z něho vycházel můj předpoklad, že i četnost výskytu znaku na dané pozici bude u Confickeru rozložena rovnoměrně. Tudíž znaky u legitimních

domén na dané pozici málo se vyskytující budou mít vyšší pravděpodobnost výskytu na oné pozici u škodlivých domén.



Obrázek 5.2: Četnost výskytu prvního znaku u Confickeru

Předpoklad o rovnoměrném rozložení četnosti znaků na určité pozici jsem ověřila předáním vzorku Confickeru navrženému systému jako modelu.

- **Kontrola prvního a posledního bigramu** je pak analogií předchozí kontroly prováděnou nad bigramy.
- **Kontrola váhy a mediánu frekvenční analýzy**

Zavedení kontroly podle mediánu vychází opět z rovnoměrného rozložení četnosti znaků u Confickeru. Tato vlastnost je sice zárukou toho, že nastavením dostatečně vysokého prahu bude výpočtem váhy frekvenční analýzy odhalena většina škodlivých domén, ale je jasné, že přes takový práh neprojde ani velké množství legitimních domén. To mě přivedlo na myšlenku odlišit tyto dvě skupiny podle mediánu. Ten najde uplatnění zejména u některých kratších legitimních domén, kdy je jejich váha snižována jedním znakem velmi nízké četnosti. Jejich medián ovšem bude podstatně vyšší než je tomu u škodlivých domén, ve kterých mají teoreticky stejné zastoupení málo početné i více početné znaky.

- **Kontrola poměru číslic ku všem znakům**
- **Kontrola poměru samohlásek ku všem písmenům**
- **Kontrola délky domény**

6 Implementace

Navržený program je implementován v jazyce C++ a skládá se ze dvou hlavních částí, a to z části pro vytvoření modelu z předaného seznamu doménových jmen a části pro detekci škodlivých domén v zachyceném DNS provozu za použití vytvořeného modelu.

6.1 Implementace vytvoření modelu

Motivací pro implementaci této části byla možnost získat automatizovaně z libovolného seznamu doménových jmen model, oproti kterému pak jsou testována doménová jména získaná z reálného provozu. Vedlejším produktem je vytvoření souborů, ze kterých lze získat statitistiky o různých parametrech doménových jmen.

Vstupem pro tuto část je textový soubor, který obsahuje na každém řádku jedno doménové jméno. Pro co nejlepší výsledky by takový soubor měl obsahovat ideálně co nejvíce doménových jmen. Na druhou stranu je potřeba počítat s tím, že program v této části pracuje s velkými datovými strukturami a tudíž by měla být velikost předaného souboru přiměřená. Například v případě seznamu od Alexy, který byl použit jako model pro testování programu, se jednalo o soubor o velikosti cca 16,3 kB.

6.1.1 Filtrace doménových jmen

Použitý seznam od Alexy byl také příčinou implementace určitého typu filtrace předaných doménových jmen. V tomto seznamu lze totiž najít doménová jména i s navigací webové stránky, doménová jména s kódováním UNICODE, hashe se znaky, které nejsou povoleny pro tvoření doménových jmen, a dokonce i IP adresy. Proto je potřeba provést před samotným získáním modelu filtraci, která během průchodu předaného souboru doménových jmen zahrnuje následující kroky:

- Ořezání doménového jména od navigace stránky. Např. ze zhidao.baidu.com/user/admin se stane zhidao.baidu.com.
- Odstranění doménových jmen s UNICODE kódováním, tedy se znaky, které nejsou v základu povoleny pro tvorbu doménových jmen. Analýza těchto doménových jmen není z důvodu zjednodušení implementována. Takové doménové jméno se pozná podle řetězce “xn--”, který je na začátku některé jeho domény. Např. xn--frstrowsports-39b.eu, kde hexidecimální hodnota 39b značí znak řecké abecedy ‘Λ‘.
- Odstranění řetězců, které neobsahují žádnou tečku, tedy rozhodně nejsou doménovými jmény.
- Odstranění IP adres je provedeno kontrolou TLD. Pokud je to číslo, nemůže to být doménové jméno. Použití číslic pro TLD sice není zakázáno, ale žádná taková veřejná doména zatím neexistuje.
- Odstranění duplicitních doménových jmen získaných zejména ořezáním od navigace stránky. Např. ve verzi seznamu použité pro testování se vyskytuje víc jak 1 000 odkazů na různé kanály uživatelů youtube.com. Pro další zpracování se ale doménová jména, která zatím prošla filtrací, ukládají do struktury, jejíž položky jsou unikátní, takže youtube.com je v ní ve výsledku zastoupeno jen jednou.

Do struktury stejného typu se ukládají také všechny TLD. Vzniklé struktury jsou pak použity v kroku, který následuje po průchodu celého souboru a jeho uzavření.

V tomto kroku dochází k filtraci jednotlivých domén a to opět za pomoci struktury unikátních hodnot. Myšlenka je taková, že např. doména google, která je v daném seznamu zastoupena s několika různými TLD, se opět uloží jen jednou. To stejné platí pro často užívanou doménu nejnižší úrovně www. Zároveň je ovšem nežádoucí, aby se ukládaly i TLD. V základě by se tohoto cíle dalo snadno dosáhnout zjištěním počtu domén v doménovém jméně a ukončením zpracování doménového jména, jakmile by nějaký čítač dosáhl tohoto počtu, což by indikovalo, že ke zpracování zbyla poslední doména. Existují však i ustálené kombinace několika TLD jako např. co.uk. V takovém případě by se program pokusil o uložení domény co. Z tohoto důvodu je implementováno řešení, kdy je každá doména testována na přítomnost v dříve získané struktuře TLD. Pokud je v ní obsažena, uloží se pouze, pokud je doménou nejnižší úrovně, tedy čítač domén je na začátku.

Po naplnění struktury unikátních názvů domén následuje zpracování obsažených domén znak po znaku. V této části jsou pro účely vytvoření modelu počítány parametry jako počet číslic, samohlásek či pomlček v doméně a díky zjišťování příslušnosti znaku k některé z těchto skupin, dochází také k poslední filtraci případných domén s nepovolenými znaky, např. 8666344164372770322_86e07bf2272156193ebcf70a1d3819465e6a171a.

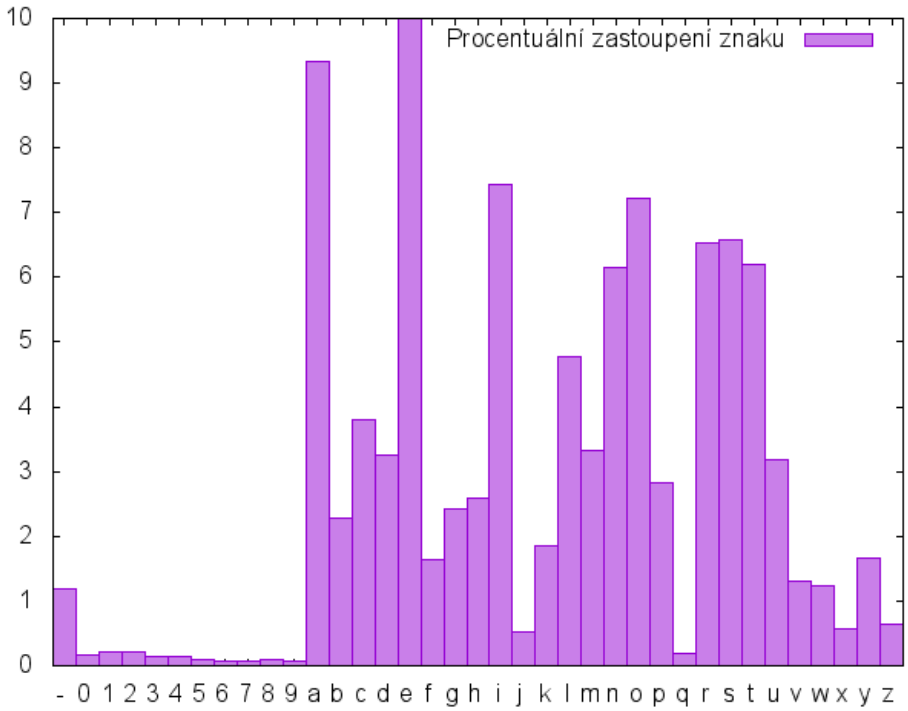
Výsledkem filtrace použitého testovacího seznamu 1 000 000 doménových jmen od Alexy je přibližně 900 000 domén.

6.1.2 Model

Výsledný model se skládá ze dvou částí, a to z frekvenční analýzy a z mezních hodnot. Obě tyto části jsou použity v následné analýze provozu DNS. Frekvenční analýza sestává z četností znaků vyskytujících se v předaném seznamu doménových jmen. Jednotlivé četnosti se počítají podle vzorce:

```
četnost v % = (počet výskytů znaku / suma výskytů všech znaků) * 100.0
```

Z testovaného seznamu vzešlo následující rozložení četnosti znaků:



Obrázek 6.1: Frekvenční analýza získaná ze seznamu Alexa

Mezní hodnoty jsou různého typu a způsob jejich získání se liší. Pro váhu a medián frekvenční analýzy byl implementován výpočet mezní hodnoty podle vzorce:

```
mezní hodnota = 100.0 / počet vyskytujících se znaků
```

Tento vzorec odpovídá rovnoměrnému rozložení všech zastoupených znaků a výsledky se liší právě podle počtu různých znaků obsažených v předaném seznamu doménových jmen. Pokud se vyskytují všechny použitelné znaky (písmena, číslice, pomlčka), je výsledkem hodnota `100.0 / 37 = 2.70`. Pokud by ale frekvenční analýzu tvořila jen písmena, byl by výsledek `100.0 / 26 = 3.85`.

Z obrázku 6.1 je jasně patrné, že číslice mají v doménových jménech suverénně nejnižší četnost a z písmen se jim v tomto ohledu blíží jen znak ‘q’. Toto pozorování jsem uplatnila během implementace testu na váhu a medián frekvenční analýzy domény. Pokud bych za mezní hodnotu zvolila hodnotu vypočtenou z počtu všech použitelných znaků, drtivá většina domén neobsahujících číslo by přes tuto hranici přešla a to včetně těch škodlivých. Např. doména `txkjngucnth` by měla váhu `3.59`. Naopak pokud by byla mezní hodnotou ta, která vzešla z počtu všech vyskytujících se písmen, uvedená doména by už tímto testem neprošla. S touto hodnotou by ovšem mohli mít problém zase některé legální domény obsahující číslice, zejména pak ty kratší jako např. `9gag`, jejíž váha je `3.56`. Proto jsem se rozhodla pro vytvoření dvou mezních hodnot. Jednu pro domény s číslicemi a pomlčkami, druhou pro domény složené jen z písmen.

Analogie uvedených vzorců se používá také pro výpočet četností a mezních hodnot pro testovaní prvního znaku, posledního znaku, prvního bigramu a posledního bigramu. Obecně lze tak tyto dva vzorce zapsat jako:

```
četnost v % = (počet výskytů prvku v množině / suma výskytů všech prvků v množině) * 100.0
```

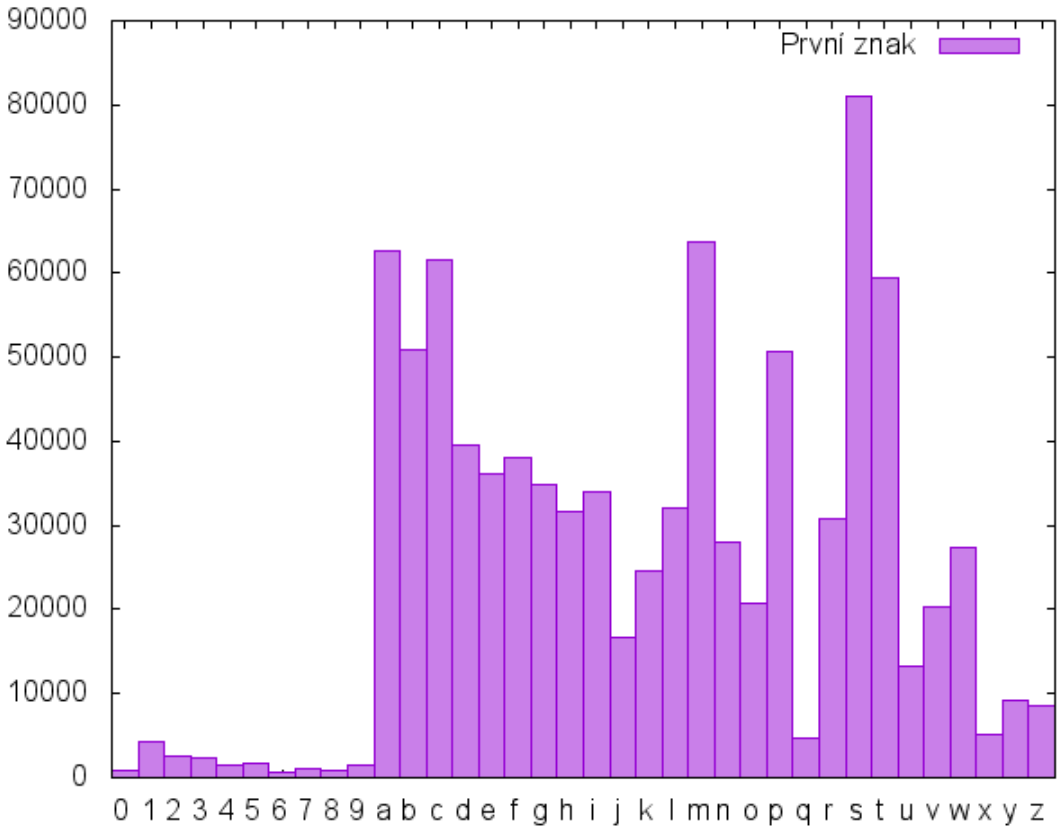
```
mezní hodnota = 100.0 / počet různých prvků množiny
```

Pokud ze zpracování použitého seznamu doménových jmen vzejdou všechny možné prvky těchto množin, pak budou mezními hodnotami:

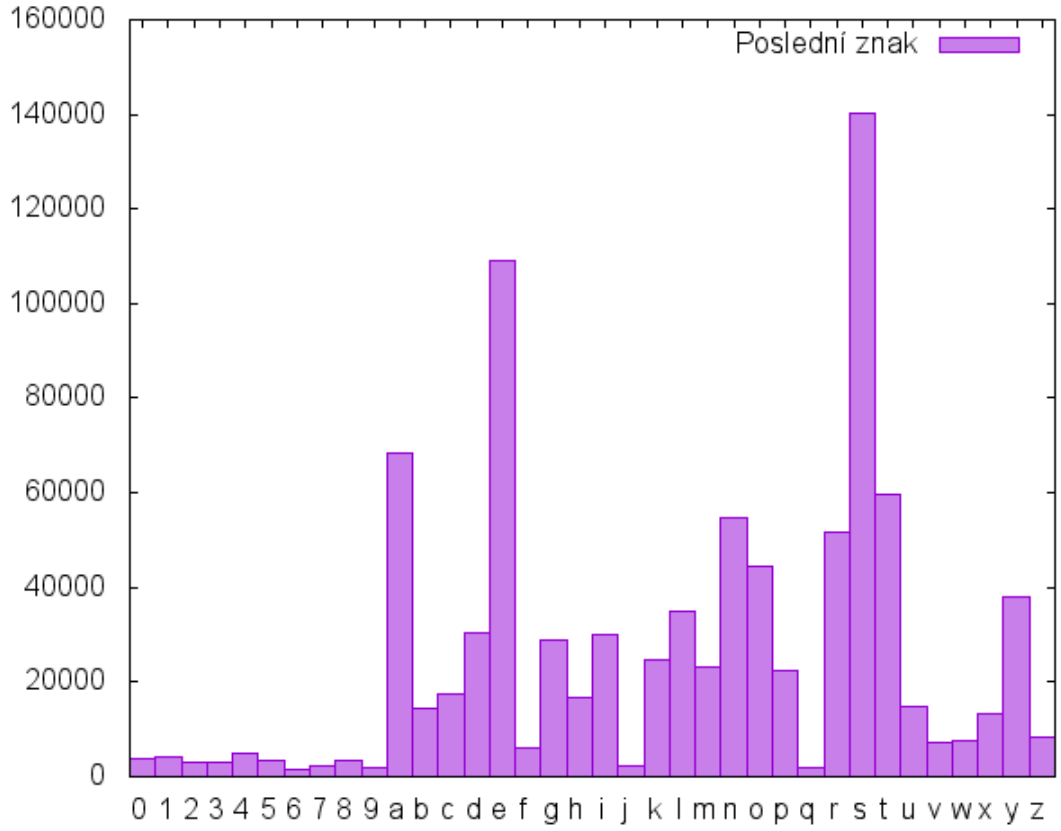
- `100 / 36 = 2.77` pro první a poslední znak (pomlčka nemůže být na začátku, ani na konci domény).
- `100 / (36*37) = 0.075` pro první a poslední bigram (druhým nebo předposledním znakem už pomlčka být může).

Tyto vypočtené mezní hodnoty ovšem v tomto případě nejsou výsledkem předaným modelu. Výsledkem je sada všech prvků, které mají nižší četnost, než je vypočtená mez. Tímto způsobem je zaručeno, že během analýzy domény už stačí jen ověřit výskyt odpovídajícího prvku v sadě, místo ověřování jeho četnosti oproti mezní hodnotě.

Nejčastější kombinací prvního a posledního znaku v použitém testovacím seznamu je dvojice znaků ‘ss’. Nejčastějším prvním bigramem je ‘ma’. Nejčastějším posledním bigramem je ‘er’. Výskyt posledního a prvního znaku v testovaném seznamu znázorňují následující grafy:



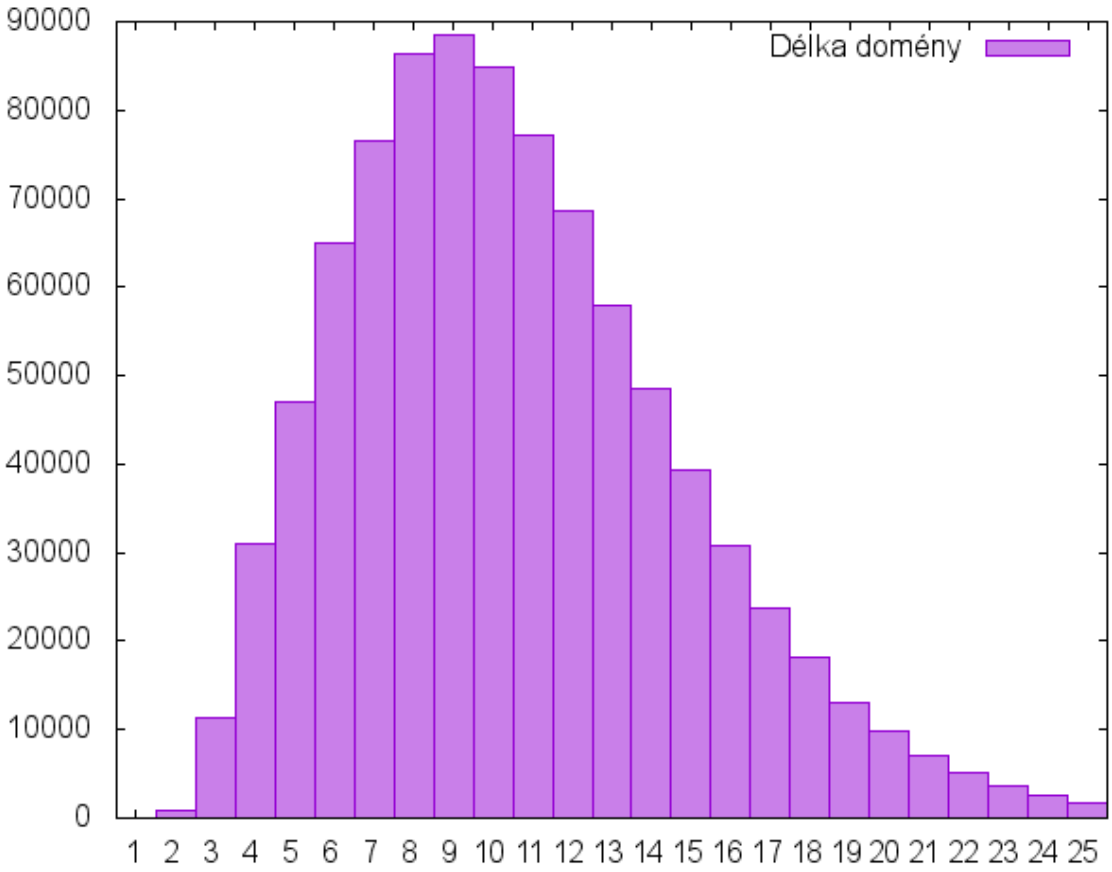
Obrázek 6.2: Rozložení výskytu prvního znaku u seznamu Alexa



Obrázek 6.3: Rozložení výskytu posledního znaku u seznamu Alexa

I pro určení mezní délky domény se používá mezní hodnota vypočtená z počtu v předaném seznamu doménových jmen zastoupených délek. Doména může mít maximálně 63 znaků, ovšem i v tak velkém vzorku doménových jmen, jako je seznam od Alexy, se nevyskytují všechny možné délky. Konkrétně v použité verzi se nachází 53 různých délek. U tohoto parametru je tak nejlépe vidět možnost proměnlivosti mezních hodnot vypočtených z různých seznamů doménových jmen.

Výslednou mezí pro tento parametr je pak co možná nejmenší délka domény, která má četnost nižší, než je vypočtená mezní hodnota. Při velkém počtu předaných doménových jmen ale vždy odpovídá taková nejmenší délka hodnotě 1, protože kvůli dříve zmíněné struktuře unikátních domén může být zpracovaných domén s touto délkou maximálně 36. Přitom je ale jisté , že se po této hodnotě nacházejí délky s mnohem vyšší četností, které by měly být větší než stanovená mez. Proto je nejprve nalezena právě délka s nejvyšší četností a až od ní výše se hledá délka, která neprojde přes danou mez. V případě použitého testovacího seznamu byla programem vypočtena mez o hodnotě 19.



Obrázek 6.4: Rozložení délky domény u seznamu Alexa

Posledními parametry domény, pro něž jsou počítány mezní hodnoty, jsou poměr číslic ku všem znakům a poměr samohlásek ku všem písmenům. V tomto případě se jako mezní hodnota používá průměr všech hodnot získaných z domén, které splňují určitá kritéria. Stejně jako u frekvenční analýzy je totiž i v případě poměru číslic ku všem znakům zohledněna skutečnost, že domén s alespoň jednou číslicí je výrazně méně, než domén bez žádné číslice. Pokud by se tedy průměr počítal ze všech domén, velmi často by se do něj připočítávala 0 reprezentující doménu bez číslice, která by snižovala celkový průměr, a výsledek by tak byl rapidně nižší, než když se tento průměr počítá jen z domén, které obsahují alespoň jednu číslici. U použitého seznamu doménových jmen se konkrétně jedná o hodnotu `0.02` pro všechny domény oproti hodnotě `0.29` pro domény obsahující číslice, což je skutečně velký rozdíl.

Pro domény určené pro výpočet poměru samohlásek ku všem písmenům platí jen jedno omezení a to, že musí obsahovat alespoň jedno písmeno. Jinak by docházelo k nepovolené operaci dělení nulou. Pro použitý seznam doménových jmen je pak vypočtenou mezí hodnota `0.39`.

6.1.3 Výstupní soubory

Výstupem této části programu je několik souborů. Prvním souborem, který je tvořen už během procházení jednotlivých domén znak po znaku, je soubor ve formátu CSV, jehož položkami jsou jednotlivé domény použité pro vytvoření modelu. Tyto domény jsou popsány následujícími atributy:

- DOMAIN_NAME doménové jméno, ze kterého vzešla zkoumaná doména
- DOMAIN_CNT počet domén
- TLD doména nejvyšší úrovně
- DOMAIN zkoumaná domén
- DOMAIN_LENGTH délka domény
- LETTER_CNT počet písmen
- VOWEL_CNT počet samohlásek
- CONSONANT_CNT počet souhlásek
- NUMBER_CNT počet číslic
- HYPHEN_CNT počet pomlček
- FIRST_CHAR první znak
- LAST_CHAR poslední znak
- FIRST_BIGRAM první bigram
- LAST_BIGRAM poslední bigram

Tento soubor je určen pro případnou důkladnější analýzu např. pomocí programu Microsoft Excel.

Další vytvořené soubory jsou statistikou o jednom konkrétním parametru domény a jsou ve formátu umožňujícím zpracování utilitou gnuplot. Tyto statistické soubory jsou k dispozici pro parametry první znak, poslední znak, první bigram, poslední bigram a délka domény, a také pro počet domén v doménovém jméně a výskyt jednotlivých TLD.

V neposlední řadě jsou vytvořeny dva soubory pro uložení vytvořeného modelu, který tak může být znovu načten při opětovném spuštění programu. Soubor, ve kterém je uložena získaná frekvenční analýza, je také ve formátu umožňujícím jeho zpracování utilitou gnuplot. Soubor s mezními hodnotami z tohoto formátu také vychází, ale protože obsahuje různé typy výsledků, z nichž některé jsou navíc tvořeny sadami hodnot, gnuplot jej nezpracuje a tento formát tak slouží spíše pro snadnou implementaci načítání modelu.

6.2 Implementace detektoru škodlivých domén

V této části programu je implementován samotný detektor škodlivých domén, který spouští analýzu doménových jmen. Vstupem je soubor ve formátu CSV s daty získanými z reálného provozu, který byl blíže popsán v kapitole 5. Aby mohla být analýza spuštěna, musí být na prvním řádku tohoto souboru uveden popis předaných dat, tedy přesněji řečeno výčet jejich atributů. Díky uvedení tohoto popisu není programem vyžadován pevně daný počet a pořadí atributů. Analýza tak může být provedena nad souborem obsahujícím všechny atributy zmíněné v kapitole 5, stejně jako nad souborem, který obsahuje položky pouze s atributem DNS_NAME.

Na začátku této části je tedy analyzován první řádek předaného datového souboru a podle přítomnosti či nepřítomnosti uvedených atributů, je rozhodnuto, jestli analýza může být spuštěna. Zároveň je také dalšímu běhu programu předáno umístění vyžadovaných atributů v položce.

6.2.1 Filtrace doménových jmen

I v této části musí probíhat určitý typ filtrace doménových jmen získaných z atributu DNS_NAME. Pro ten totiž rozhodně není zaručeno, že skutečně bude obsahovat řetězec odpovídající doménovému jménu.

- První forma filtrace může proběhnout na základě zjištění hodnoty atributu DNS_QTYPE a jeho porovnání s hodnotami odpovídajícími záznamům A a AAAA (tedy 1 a 28), čímž se odstraní záznamy jiných typů. Pokud ale tento atribut není zahrnut v předaném datovém souboru, tento krok se přeskočí.
- Obdobně může také proběhnout filtrace na základě DNS_RCODE, kdy se filtrují hodnoty NXDOMAIN (3) od ostatních.
- Zcela odstraněny jsou doménová jména s kódováním UNICODE, protože ty nejsou zahrnuty do tvorby modelu a tudíž nelze počítat s tím, že jejich analýza by měla relevantní výsledky.
- Dotazované doménové jméno může obsahovat také prázdný řetězec, který označuje doménu *root*. Takové položky jsou též odstraněny.
- Vyskytovat se mohou také řetězce neobsahující tečku. I ty nejsou do analýzy zahrnuty.
- Dalším případem jsou pak řetězce, které sice obsahují tečky, ale mají jednu přímo na začátku nebo teoreticky více po sobě, např. .sh.cvut.cz. Z pohledu programu se v takovém doménovém jméně vyskytují prázdné domény, což není povoleno.
- Posledním krokem filtrace je vyřazení domén s nepovolenými znaky.

6.2.2 Whitelist

Během implementace této části programu brzy vzešel požadavek na vytvoření whitelistu, tedy seznamu domén, které jsou považovány za důvěryhodné a projdou detektorem bez analýzy. To z toho z důvodu, že předaná data obsahovala vysoký počet antispamových kontrol a další doménová jména vzniklá nějakým automatizovaným nástrojem. Příkladem jsou:

```
10.121.206.195.zen.spamhaus.org
gobbles-aws-226538871.eu-west-1.elb.amazonaws.com
8d96Ea375A3d465fe7f7d9Db9ECe8955.IxhASh.spAmEaTiNGMoNKey.NET
```

Jak je vidět, taková doménová jména by určitě analýzou skladby doménového jména neprošla. Proto bylo v programu implementováno načítání whitelistu z externího souboru, který může být upravován podle potřeb uživatele. Je nutno ovšem podotknout, že není v této implementaci počítáno s tím, že tento whitelist bude rozsáhlejšího charakteru.

6.2.3 Analýza skladby doménového jména

Doménové jméno předané k této analýze je nejprve rozděleno na jednotlivé domény. Nezpracovávají se TLD a také domény s menší délkou než 3 znaky. Jednoznakové domény totiž nemohou projít kontrolami prvního a posledního bigramu a u dvouznakových by se těmto kontrolám zase předala stejná data. Navíc u takhle krátkých domén platí, že většinou nemohou představovat slova nějakého přirozeného jazyka a tudíž je vysoce pravděpodobné, že představují nějakou zkratku. Tříznakové domény už mohou tvořit slova přirozeného jazyka, ačkoli výskyt zkratek bude v jejich případě zřejmě také vysoký a to už jen z důvodu častého použití domén nejnižší úrovně www či ns[x] (označuje *nameserver* číslo x).

Pokud tedy doména projde všemi filtracemi, je přistoupeno k samotné kontrole jejích parametrů. K získání všech potřebných údajů o doméně je použita funkce již implementovaná v části vytvářející model. U většiny kontrol už tak stačí jen porovnat mezní hodnoty s právě získanými. Výjimkou jsou kontroly na základě frekvenční analýzy, pro které je nejen nutné vypočítat zkoumané hodnoty, ale i rozhnodnout, která mez se použije. Tato skutečnost už byla popsaná v kapitole 6.1.2.

6.2.3.1 Skóre domény

Pro zařazení domény mezi škodlivé nebo legitimní domény je použit určitý typ skóre. To je na začátku zpracování rovno 0. Pokud doména neprojde nějakou kontrolou, je od jejího skóre odečtena 1. Pokud projde, pak je 1 ve většině případů přičtena. Výjimkami jsou:

- Kontrola délky domény, v jejímž případě je předpoklad, že délka domény nepřesahující stanovenou mez je znakem legitimní domény, evidentně mylný.
- Kontrola poměru číslic ku všem znakům, u které je speciálním případem, když doména žádnou číslici neobsahuje. Rozhodnout v tomto případě, že doména neobsahující číslo je spíše legitimní, je také nevhodné. U poměru samohlásek vůči ostatním písmenům existuje také možnost, že doména neobsahuje žádné písmeno. V takovém případě je ale ze zřejmých důvodů rozhodnuto, že tato skutečnost je znakem škodlivé domény.

Tabulka 6.1 podrobně popisuje rozhodování o výstupu jednotlivých kontrol do skóre domény.

Kontrola	< mez	> mez	Chybí vstup
první znak	**-1**	**1**	**X**
poslední znak	**-1**	**1**	**X**
první bigram	**-1**	**1**	**X**
poslední bigram	**-1**	**1**	**X**
váha	**-1**	**1**	**X**
medián	**-1**	**1**	**X**
samohlásky : písmena	**-1**	**1**	**-1**
číslice : znaky	**1**	**-1**	**0**
délka domény	**0**	**-1**	**X**

Tabulka 6.1: Výstup jednotlivých kontrol analýzy skladby doménového jména

6.2.3.2 Skóre doménového jména

Po výpočtu skóre domén je spočítáno také skóre celého doménového jména (resp. všech jeho kontrolovaných domén). Jednoduchou možností by bylo jednotlivá skóre sečíst dohromady. Lepší je však zohlednit délku jednotlivých domén. Například u domény www je vysoká pravděpodobnost, že bude označena za škodlivou. V takovém případě je ovšem nežádoucí, aby se tato skutečnost výrazně promítla na celé doménové jméno, např. www.seznam.cz.

Otázkou také je, jestli je vhodné do celého skóre nadále promítat skóre jednotlivých domén. Tato otázka se opět týká kratších domén, které mají potenciál jednoduše dosáhnout skóre až -8. Na druhou stranu škály je ovšem mnohem obtížnější se dostat. Stačí, aby legitimní doména neprošla například kontrolou jednoho z unigramů a jednoho z bigramů a už má skóre 3. Pokud má tedy dvojnásobnou délku než ona problémová doména, byla by výsledkem výpočtu kombinujícího skóre s délkou domény záporná hodnota.

Proto je v navrženém programu implementován vzorec, který podle skóre přiřadí doméně opět hodnotu 1 nebo -1 a těmi pak násobí délku domény.

```
celkové skóre = 1 * počet znaků legitimních domén +
                (-1) * počet znaků škodlivých domén
```

Pokud je celkové skóre menší než 0, je doménové jméno označeno jako škodlivé. Určení toho, jaké skóre domény by mělo být hraničním, je pak jedním z cílů testování.

6.2.4 Výstupní soubory

Výstupem této části jsou textové soubory popisující výsledky provedených kontrol. Pro každou kontrolu je vytvořen zvláštní soubor, ve kterém je na prvním řádku uveden celkový počet domén, které touto kontrolou neprošly a na druhém procentuální vyjádření jejich zastoupení mezi všemi kontrolovanými doménami. Následuje výpis všech detekovaných domén. Podobný soubor je také vytvořen pro celkový výsledek detektoru, tedy jeho obsahem je výpis všech doménových jmen, které byly označeny jako příslušející nějaké škodlivé doméně.

K dispozici je také soubor s detailním výpisem průběhu analýzy. U každého analyzovaného doménového jména je uveden seznam jeho testovaných domén s podrobnými výsledky jednotlivých kontrol a také vypočtené celkové skóre domény. Např.:

```
dhs.gov
    dhs,1,1,-1,-1,1,-1,-1,0,0,-1
SCORE:     -3
ipnp00.troja.mff.cuni.cz
    ipnp00,1,-1,1,1,1,1,-1,-1,0,2
    troja,1,1,1,1,1,1,1,0,0,7
    mff,1,-1,-1,1,-1,-1,-1,0,0,-3
    cuni,1,1,1,1,1,1,1,0,0,7
SCORE:     12
    ns.kiev.farlep.net
    ,0
    kiev,-1,-1,1,1,1,1,1,0,0,3
    farlep,1,-1,1,1,1,1,-1,0,0,3
SCORE:     10
```

7 Testování

Jako modelový soubor byl během testování používán seznam doménových jmen od Alexy. Důležitým referenčním datovým souborem pak byl vzorek doménových jmen Confickeru, u kterého bylo cílem dosáhnout co největší míry detekovaných domén. Opačného efektu pak bylo účelem docílit u volně dostupného seznamu všech doménových jmem domény `sk`, u kterého se dá předpokládat, že je tvořen legitimními doménami. Soubory dat získaných z reálného provozu byly rozděleny na menší a typově se dají rozdělit na tři skupiny podle zastoupení atributů používaných během analýzy. Parametry souborů použitých pro následné hodnocení jsou tyto:

Soubor	Počet položek	Velikost
alexa	1 000 000	16,4 MB
conficker	113 500	1,6 MB
domainsk	315 083	4,6 MB
dnsname	1 000 000	22,9 MB
qtype	1 000 000	25,5 MB
rcode	200 000	32,4 MB

Tabulka 7.1: Přehled souborů použitých při testování

Soubor dnsname obsahuje položky pouze s atributem DNS_NAME, u qtype je přítomen i DNS_QTYPE a analogicky u rcode DNS_RCODE. U posledně zmiňovaného jsou pak dokonce zastoupeny i všechny ostatní možné atributy a tomu také odpovídá poměr jeho velikosti vůči počtu položek. V tabulce 7.2 je pak možné sledovat, jak moc byla velká míra filtrace těchto souborů, přičemž u souboru rcode (projde kombinace záznamu typu A nebo AAAA a NXDOMAIN) se jednalo o velmi razantní pokles. Tabulka také ukazuje výsledky jednotlivých kontrol detektoru škodlivých domén při základním nastavení limitního skóre domény na hodnotu 0.

Výsledkem této analýzy je poznatek, že pro domainsk se skutečně podařilo dosáhnout poměrně nízké míry detekce. Lze však takové hodnoty dosáhnout i u reálného provozu? Pro nalezení odpovědi na tuto otázku je nutno zaměřit se na to, jaké kontroly mají značně rozdílný výsledek. Těmi jsou v tomto případě obě kontroly založené na frekvenční analýze a kontrola poměru číslic ku všem znakům. Co tedy způsobuje tak velký rozdíl? U kontrol založených na frekvenční analýze padlo podezření na již dříve zmiňované krátké domény typu www. Soubor domainsk totiž obsahuje pouze domény druhé úrovně, a tak je pravděpodobnost, že se v něm vyskytne taková problémová doména, nižší než u mnohonásobně zanořených domén, které lze nalézt v souborech s reálnými daty. Pro ověření tohoto podezření bylo tedy filtrování domén krátkodobě zvýšeno i na trojznakové domény. Výsledky tohoto opatření jsou zobrazeny v tabulce 7.3.

detekce	conficker	domainsk	dnsname	qtype	rcode
položky	113 500	315 083	976 381	764 524	4 124
první znak	34,72 %	18,78 %	15,16 %	17,67 %	27,91 %
poslední znak	53,91 %	23,51 %	41,95 %	48,25 %	44,57 %
první bigram	70,35 %	14,64 %	34,43 %	40,29 %	33,49 %
poslední bigram	67,81 %	9,65 %	36,33 %	44,97 %	45,37 %
váha	55,53 %	2,82 %	17,29 %	26,21 %	26,64 %
medián	77,53 %	11,71 %	29,81 %	37,12 %	39,19 %
samohlásky / písmena	84,51 %	42,14 %	62,98 %	63,94 %	64,04 %
číslice / znaky	0 %	0,77 %	12,62 %	13,82 %	17,53 %
délka	0 %	2,39 %	1,01%	0,80 %	3,40 %
celkový výstup	71,24 %	6,00%	14,59 %	23,43 %	14,06 %

Tabulka 7.2: Výstup jednotlivých částí detektoru pro různé soubory

detekce	domainsk	dnsname	qtype	rcode
váha	1,95 %	8,95 %	19,11 %	11,30 %
medián	10,53 %	18,27 %	28,71 %	20,81 %

Tabulka 7.3: Snížení míry detekce váhy a mediánu frekvenční analýzy po vyřazení trojznakových domén

Je vidět, že u souborů s reálnými daty skutečně došlo ke snížení výskytu této detekce i více než o polovinu, zatímco u domainsk už se tyto hodnoty příliš nepohybovaly. Podobně i vyšší výskyt detekce kontroly poměru číslic ke všem znakům je zřejmě způsobena výskytem mnohaúrovňových doménových jmen.

O dříve uvedeném požadavku na nízký výskyt detekce u domainsk se tedy dá prohlásit, že byl splněn. U souboru conficker je však splnění vzneseného požadavku diskutabilní. Proto jsem provedla test, ve kterém jsem navýšila limitní skóre domény tak, aby výskyt detekce u souboru conficker přesáhl 90 %. Toho jsem dosáhla u hodnoty 4. Je nutno podotknout, že tímto krokem se výstupy jednotlivých kontrol nezmění. Mění se jen konečný výsledek a cílem tohoto testu tak bylo zjistit, jakým způsobem narůstá výskyt detekce u jednotlivých souborů, tedy jestli je tento průběh obdobný nebo se liší. Výsledek je zobrazen v tabulce 7.4.

detekce	conficker	domainsk	dnsname	qtype	rcode
celkový výstup	94,42 %	32,55 %	34,81 %	45,46 %	32,13 %

Tabulka 7.4: Celkový výstup po navýšení limitního skóre domény

U souborů s daty z reálného provozu došlo k navýšení výskytu detekce zhruba o dvojnásobek. U domainsk pak dokonce o pětinásobek, takže je evidentní, že takové pohyby se skórem domény mají nevhodný dopad na celkový výsledek.

8 Závěr

Výsledkem této práce je systém pro detekci škodlivých domén v provozu DNS. Aby takový systém mohl být navržen a implementován, bylo nejdříve potřeba pochopit to, proč je vlastně tato problematika tak důležitá. Důležitost systému DNS tedy byla načrtnuta už v úvodě a druhá kapitola se podrobně věnovala jeho fungování. Tím byla také naznačena značná šíře tématu věnujícího se bezpečnosti DNS, kterou tato bakalářská práce rozhodně nemůže pokrýt celou.

Proto byla ve třetí kapitole představena konkrétní bezpečnostní hrozba vyskytující se v provozu DNS, a tou je působení ilegálních botnetů využívajících DGA. Stopy, které tyto botnety za sebou zanechávají, pak vedly k návrhu samotného systému. Ještě předtím však bylo nutné ujasnit si, jak se k těmto stopám dostat, a tomu se věnuje čtvrtá kapitola o monitorování síťového provozu.

Návrh systému pak byl vystavěn na atypické skladbě doménových jmen generovaných DGA. Aby však bylo možné odlišit škodlivá doménová jména od těch legitimních, vyvstala nutnost použít pro toto rozhodování nějaký model existujících doménových jmen. Hledání vhodného modelu pak vedlo k tomu, že ve výsledném systému je implementován postup vytvoření modelu z jakéhokoli seznamu doménových jmen. Implementace samotné detekce se pak skládá z řady jednotlivých kontrol, z jejichž výsledků je pak určen výsledek celkový.

Pomocí vzniklého systému je tak možné zkoumat vhodnost zvolených modelů a právě v rozšíření tohoto výzkumu vidím možnost pokračování této práce. Zkoumáním dopadu volby určitého modelu na jednotlivé představené kontroly by mohl vzniknout mnohem komplexnější systém pro analýzu skladby doménových jmen.

Zajímavou by také byla jistě diskuze o tom, jak počítat celkové skóre doménového jména. Např. měly by mít všechny uvedené kontroly stejnou váhu? Prostor pro řešení této otázky a dalších, které mohou napadnout někoho jiného, je volný.

Literatura

[1] MATOUŠEK, Petr. *Síťové aplikace a jejich architektura*. Brno: VUTIUM, 2014. ISBN 978-80-214-3766-1.

[2] Verisign: March 2015: Domain Name Industry Brief. [online]. [cit. 2015-05-18]. Dostupné z: http://www.verisigninc.com/assets/domain-name-report-march2015.pdf

[3] CZ.NIC: Domain Report 2014. [online]. 2015-02-02 [cit. 2015-05-18]. Dostupné z: https://stats.nic.cz/reports/2014/

[4] YUE, Frank: DNS for Communication Service Providers Part III - Security. [online]. 2013-04-03 [cit. 2015-05-18]. Dostupné z: https://devcentral.f5.com/articles/dns-for-communications-service-providers-part-iii-security

[5] Damballa: DGAs in the Hands of Cyber-Criminals: Examining the State of the Art in Malware Evasion Techniques. [online]. [cit. 2015-05-18]. Dostupné z: https://www.damballa.com/downloads/r_pubs/WP_DGAs-in-the-Hands-of-Cyber-Criminals.pdf

[6] MalwareTech: Peer-to-peer Botnets for Beginners. [online]. [cit. 2015-05-18]. Dostupné z: http://www.malwaretech.com/2013/12/peer-to-peer-botnets-for-beginners.html

[7] abuse.ch: ZeuS Gets More Sophisticated Using P2P Techniques. [online]. 2015-02-02 [cit. 2015-05-18]. Dostupné z: https://stats.nic.cz/reports/2014/

[8] DOYLE, Ryan. Frequency analysis of second-level domain names and detection of pseudo-random domain generation. [online]. 2010 [cit. 2015-05-18]. Dostupné z: http://ryandoyle.net/assets/papers/Frequency_analysis_second_level_domains_June_2010_RDoyle.pdf

[9] MAZZOCCHIO, Daniele: OpenBSD as a domain name server. [online]. 2013-11-10 [cit. 2015-05-18]. Dostupné z: http://www.kernel-panic.it/openbsd/dns/index.html

[10] Google: About DNS. [online]. [cit. 2015-05-18]. Dostupné z: https://support.google.com/domains/answer/3251148?hl=en

[11] ITgeared: How DNS Works. [online]. 2012-06-01 [cit. 2015-05-18]. Dostupné z: http://www.itgeared.com/articles/1354-domain-name-system-dns-tutorial-overview/

[12] H3C: DNS Configuration. [online]. [cit. 2015-05-18]. Dostupné z: http://www.h3c.com/portal/Technical_Support___Documents/Technical_Documents/Switches/H3C_S5500_Series_Switches/Configuration/Operation_Manual/H3C_S5500-SI_OM-Release_1205%28V1.03%29/200706/205811_1285_0.htm

[13] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan. Detecting algorithmically generated malicious domain names. In *Proceedings of the 10th annual Conference on Internet Measurement*, IMC ’10, pages 48–61, New York, NY, USA, 2010. ACM.

[14] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From throw-away traffic to bots: Detecting the rise of dga-based malware. In Proc. of the 21th USENIX Security Symposium (Security’12), Bellevue, Washington, USA, pages 48–61. USENIX Association, August 2012.

[15] Doležal Jiří: Detekce škodlivých domén za pomoci analýzy pasivního DNS provozu, diplomová práce, Brno, FIT VUT v Brně, 2014

[16] Michal Kováčik: Detekce síťových anomálií a bezpečnostních incidentů s využitím DNS dat, pojednání k tématu disertační práce, Brno, FIT VUT v Brně, 2014

[17] KOVÁČIK, Michal. Liberouter: DNS plugin. [online]. [cit. 2015-05-18]. Dostupné z: https://www.liberouter.org/technologies/dns-plugin/

Příloha A

Obsah CD

K bakalářské práci je přiložené CD, které obsahuje:

- složku src obsahující zdrojové soubory programu a ukázkové datové soubory
- soubor README pojednávající o použití programu
- výsledný text bakalářské práce “Detekce škodlivých domén pomocí analýzy DNS provozu.pdf“
- zdrojový text bakalářské práce “Detekce škodlivých domén pomocí analýzy DNS provozu.docx“