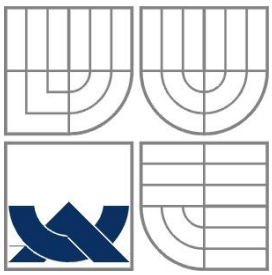
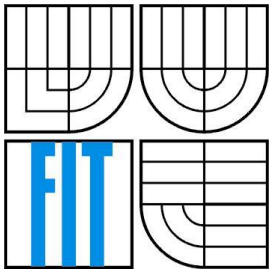


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



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

DETEKCE DNS ANOMÁLIÍ NA ZÁKLADĚ METODY PODOBNOSTI A ENTROPIE

DNS ANOMALY DETECTION BASED ON THE METHOD OF SIMILIARITY AND ENTROPY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ ŠKORPIL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL KOVÁČIK

BRNO 2014

Zadání bakalářské práce/16147/2013/xskorp05

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2013/2014

Zadání bakalářské práce

Řešitel: **Škorpil Jiří**
Obor: Informační technologie
Téma: **Detekce DNS anomálií na základě metody podobnosti a entropie**
DNS Anomaly Detection Based on the Method of Similiarity and Entropy
Kategorie: Počítačové sítě

Pokyny:

1. Seznamte se službou a protokolem DNS, nastudujte si metody detekce pomocí podobnosti a entropie.
2. Seznamte se s formáty NetFlow a pcap, následně analyzujte dodaný zachycený DNS provoz.
3. Navrhněte nástroj pro detekci anomálií DNS na základě prostudovaných metod.
4. Implementujte navržený nástroj.
5. Výsledný nástroj otestujte a zhodnoťte výsledky.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 a 2 ze zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kováčik Michal, Ing.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2013
Datum odevzdání: 21. května 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá detekcí DNS anomálií v zachyceném síťovém provozu na základě metody podobnosti a metody entropie. Cílem této práce je návrh a implementace aplikace, jež implementuje obě metody pro detekci anomálií a na základě jejich výsledků rozhodne o výskytu anomálie. Aplikace dokáže zpracovat zachycený provoz ve formátech pcap a NetFlow.

Abstract

This bachelor's thesis deals with DNS anomaly detection in captured network traffic based on the method of similarity and method of entropy. The aim of this work is design and implementation of application which implements both anomaly detection method and based on their results decides on the occurrence of anomaly. Application can handle captured traffic in pcap and NetFlow formats.

Klíčová slova

DNS, detekce anomálií, podobnost, entropie, pcap, NetFlow

Keywords

DNS, anomaly detection, similarity, entropy, pcap, NetFlow

Citace

Škorpil Jiří: Detekce DNS anomálií na základě metody podobnosti a entropie, bakalářská práce, Brno, FIT VUT v Brně, 2014

Detekce DNS anomálií na základě metody podobnosti a entropie

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Kováčika.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jiří Škorpil
21. května 2014

Poděkování

Na tomto místě bych chtěl poděkovat svému vedoucímu bakalářské práce Ing. Michalu Kováčikovi za odborné konzultace, připomínky a metodické vedení práce.

Obsah

1 Úvod

Internet je dnes již naprosto běžnou součástí života. Možnosti jeho využití jsou prakticky neomezené, pro většinu lidí však slouží nejčastěji pro komunikaci s druhými, vyhledávání informací, sdílení dat, hraní her, přístup do internetového bankovnictví, sledování videí, atd. Málokdo si ale uvědomuje důležitost systému doménových jmen (DNS), které každý zadává do adresního řádku svého internetového prohlížeče. Bez něj je totiž většina uživatelů od Internetu odříznuta. Je tedy důležité udržet tento systém v provozu a mít přehled o jeho stavu a potenciálních hrozbách.

Základním, ale nikoliv jediným, úkolem systému DNS je provádět překlad doménových jmen na IP adresy a obráceně. Služba vznikla především z důvodu zjednodušení přístupu uživatelů do sítě internet, poněvadž je mnohem snadnější zapamatovat si doménové jméno, jež je většinou vytvořeno ze samotného názvu služby, než čtveřici osmibajtových čísel (zapsáno v desítkové soustavě) tvořící IPv4 adresu, o IPv6 adrese složené z osmi skupin čtyř hexadecimálních číslic ani nemluvě. V dnešní době navíc spoustu webových služeb běží na stejné IP adrese, jednak z důvodu nedostatku IPv4 adres a také kvůli lepšímu využití fyzických zdrojů (spoustu webů nevyžaduje celý výkon serveru). Odlišení pouze pomocí IP adresy by tak nestačilo pro identifikování požadované služby a URL by musela obsahovat další údaje pro identifikaci.

Díky decentralizaci celého systému DNS jsou široké možnosti, jak zajistit bezpečnost a odolnost systému vůči útokům. Předně je tu 13 kořenových jmenných serverů, které jsou fyzicky tvořeny několika stovkami serverů rozesetými po celém světě. Na tuto klíčovou infrastrukturu bylo v minulosti spoustu útoků, ale žádný z nich neměl větší úspěch. Kořenové servery ale nejsou jediné místo, kde se dá útočit. Problémem původního návrhu systému byla snadná zneužitelnost, protože zde nebylo žádné ověření, zda data pochází z důvěryhodného zdroje (a ne od útočníka). Integritu dat nově zajišťuje rozšíření DNSSEC pomocí mechanismu podepisování záznamů. Dnes je pomocí DNSSEC podepsána celá kořenová zóna, což je důležité, protože jde o nejvyšší bod celé hierarchie podepisování. Nicméně problémem zůstává, že velká část serverů stále nepodporuje DNSSEC. Této skutečnosti může útočník zneužít. Pro posílání dotazů a odpovědí se používá nespojový protokol UDP, který nezajišťuje, že se datagram po cestě od zdroje k cíli neztratí, klient tak může být nucen dotaz poslat vícekrát. Útočníkovi stačí poslat tazateli podvrženou odpověď dříve než DNS server a má vyhráno. Až tak jednoduché to samozřejmě není, protože podvržená odpověď musí směřovat na správný port tazatele a musí mít správné identifikační číslo. Nicméně obě hodnoty jsou 16-bitové, možných kombinací je tedy pouze 2^{32} a to není tolik, aby je nešlo vyzkoušet hrubou silou. Navíc se tato čísla dají předvídat nebo hádat. A to není zdaleka jediný bezpečnostní problém, stále větší hrozbou jsou, u crackerů velmi oblíbené, útoky typu odmítnutí služby (DoS).

Cílem této bakalářské práce je na základě prostudování metod detekce DNS anomálií pomocí podobnosti a entropie navrhnout a implementovat nástroj, který analyzuje zachycený síťový provoz ve formátech pcap nebo NetFlow a detekuje v něm DNS anomálie.

Práce se skládá z několika kapitol, které postupně rozebírají jednotlivé etapy životního cyklu práce. V následující kapitole se čtenář seznámí s účelem a základním principem systému doménových jmen DNS a také protokolem NetFlow, jakožto zdrojem síťových dat. Třetí kapitola vysvětluje, co jsou to DNS anomálie, jak se projevují a jaké jsou možnosti jejich detekce. Dále pak popisuje principy metod detekce anomálií pomocí metody podobnosti a entropie. Čtvrtá kapitola se zabývá popisem návrhu nástroje. Pátá kapitola se věnuje implementaci výše zmíněného nástroje včetně popisu použitých knihoven pro čtení zdrojových dat – pcap a nfreader. Šestá kapitola popisuje způsob a výsledky testování nástroje. V závěrečné kapitole jsou shrnuty výsledky a navrhnuty možné způsoby rozšíření.

2 Systém DNS

Služba DNS (Domain Name System) představuje jednu z klíčových rolí při přístupu k celosvětové síti Internet. Její základní funkce spočívá v překladu doménových jmen na logické adresy (IP adresy), např. doménové jméno www.fit.vutbr.cz přeloží na IPv4 adresu 147.229.9.23. Při práci se sítí Internet se dnes používají doménová jména z několika důvodů, pro běžné uživatele je nejdůležitější kritériem snadná zapamatovatelnost oproti IP adresám (nemluvě o IPv6 adresách).

Z toho vyplývá, že je nutné udržet tento systém bez výpadků. V okamžiku, kdy je systém nedostupný, je uživatel prakticky odstřižen od internetu, protože si IP adresy nepamatuje, a i kdyby ano, tak spousta služeb potřebuje mít ke své činnosti přístup k systému DNS.

Kromě základní funkce překladu doménových jmen zajišťuje služba DNS i přístup k dalším datům (například různé textové údaje, adresy poštovních serverů či regulární výrazy použité pro mapování telefonního čísla na URI v IP telefonii).

2.1 Architektura

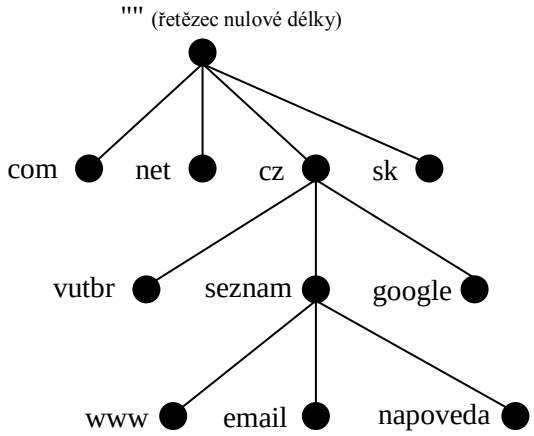
Systém DNS se skládá ze tří hlavních částí – prostoru doménových jmen, serverů DNS a rezoluce DNS dotazů prováděném tzv. resolverem.

Prostor doménových jmen

Organizace systému DNS probíhá pomocí hierarchického uspořádání záznamů do stromu (acyklický graf), viz Obrázek 1. Kořen stromu tvoří speciální záznam, jehož jméno tvoří řetězec nulové délky. Každý uzel stromu tvoří doménu v systému DNS. Listy stromu představují konkrétní zařízení v rámci dané domény. Doménové jméno představuje cestu mezi listem a kořenem stromu, kde každou část odděluje tečka. Absolutní doménové jméno tedy vždy končí tečkou, např. „`www.seznam.cz.`“ (za poslední tečkou následuje jméno kořene stromu – řetězec nulové délky). Doménové jméno, které tečku na konci nemá, označujeme jako relativní. Při vytváření DNS dotazu se za relativní doménové jméno dosadí doména, ve které se pohybujeme.

Tento postup se označuje jako tzv. přímé mapování (překlad doménového jména na IP adresu, popř. jiný údaj). Důležitou součástí systému DNS je ale také tzv. reverzní, neboli zpětné mapování, při kterém překládáme IP adresu zpět na doménové jméno. Využívá se jako kontrola, zda se někdo nesnaží podvrhnout IP adresu (DNS dotaz na doménové jméno získané zpětným překladem musí vrátit stejnou IP adresu), nejčastěji poštovními servery pro ochranu před rozesíláním spamu. Pro reverzní mapování existuje v prostoru doménových jmen speciální vyhrazená doména s názvem „`in-addr.arpa.`“. Ukládání IP adres probíhá opačně než při přímém mapování, tedy pozpátky po jednotlivých bajtech IP adresy, kde každý bajt představuje jednu úroveň stromu. Pro IPv4 adresu `77.75.76.3` bude existovat záznam `3.76.75.77.in-addr.arpa.`. Pro IPv6 adresy existuje reverzní doména „`ip6.arpa.`“. Vzhledem k délce IPv6 adresy by správa této domény byla extrémně náročná, takže existují automatizované nástroje, které se o to postarají. Tento způsob mapování umožňuje sjednotit správu jednotlivých subdomén a jim odpovídajícího prostoru IP adres.

Domény jsou spravovány decentralizovaně a tvoří logický pohled na strukturu systému DNS. Z pohledu fyzického mluvíme o tzv. DNS zónách, které jsou uloženy na DNS serverech.



Obrázek 1: Hierarchické uspořádání doménových jmen

DNS server

DNS server je aplikace, která odpovídá na DNS dotazy na základě svých DNS záznamů. Rozlišujeme několik typů DNS serverů.

Primární DNS server obsahuje kompletní DNS záznamy o doménách, které spravuje. Těmto záznamům se říká autoritativní a jsou vždy aktuální.

Sekundární DNS server získává DNS záznamy od primárního DNS serveru pomocí tzv. zónových transferů. Takto získané záznamy jsou také autoritativní.

Záložní (caching only) DNS server disponuje pouze vyrovnávací pamětí, ve které hledá odpovědi na dotazy. Pokud v ní odpověď nenalezne, předává dotaz na další DNS servery a odpověď si do ní následně uloží. Tyto záznamy ve vyrovnávací paměti zůstanou, dokud jim nevyprší platnost. Proto nemusí být aktuální a tyto odpovědi se nazývají neautoritativní.

Kořenový (root) server ukládá kořenový zónový soubor, který popisuje umístění autoritativních serverů nejvyšší úrovně. Kvůli vysokému počtu požadavků na záznamy nejvyšší úrovně existuje 13 kořenových serverů označených A-M, přičemž ani to by nezajistilo dostatečnou odolnost systému, takže fyzické servery jsou v podstatně větším počtu rozesety po celém světě. Tento systém se ukázal jako dostatečně robustní a odolal všem dosavadním útokům.

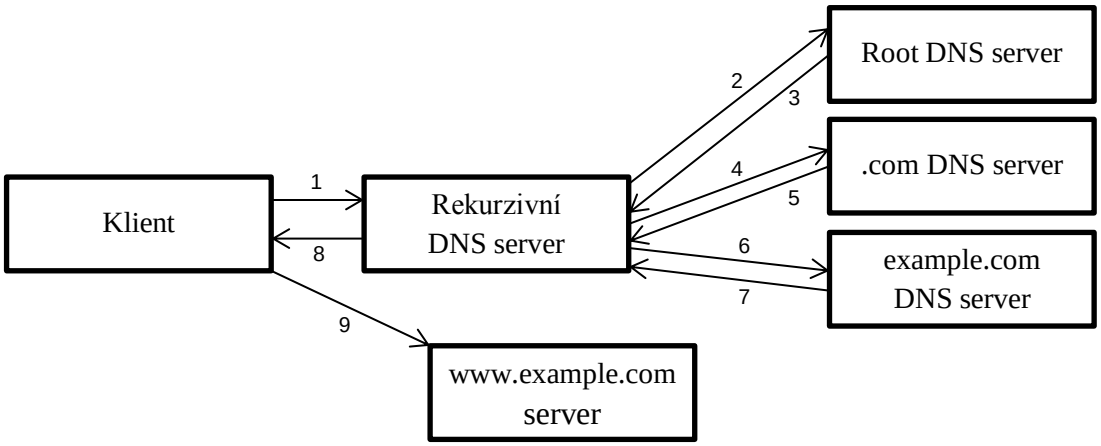
Resolver

Resolver se stará o překlad doménového jména na straně klienta. Hlavní činností resolveru je posílat DNS dotazy na servery, interpretovat jejich odpovědi a ty následně předat programu, který o překlad požádal (např. webový prohlížeč). Resolver je většinou implementován přímo v operačním systému a přistupovat k němu lze pomocí příslušeného API. Další možností je využít např. aplikaci *nslookup,* pomocí které lze jednoduše provádět DNS dotazy.

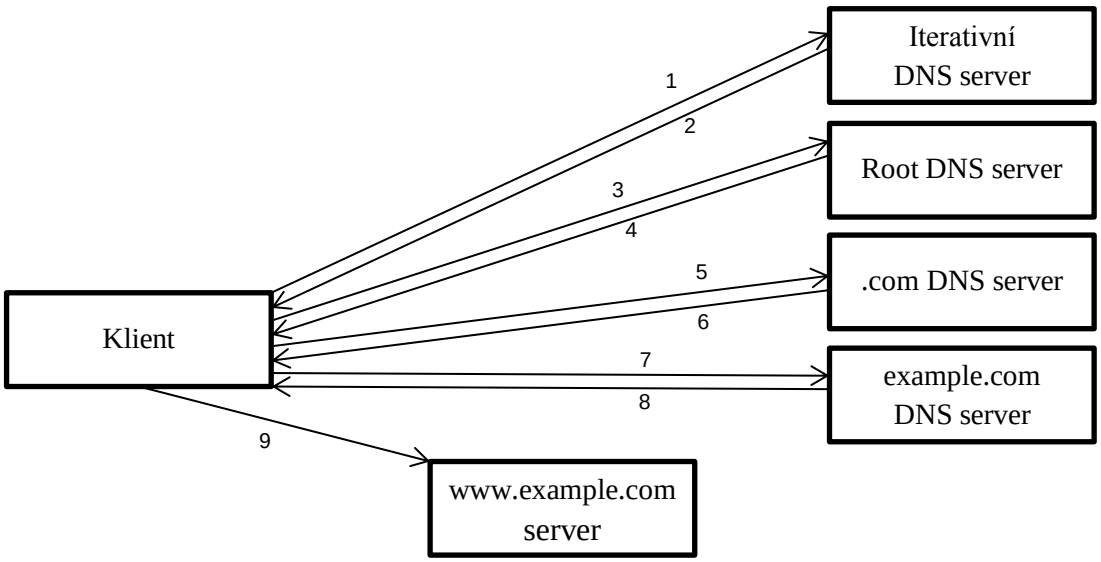
Rezoluce

Překladu doménového jména se říká rezoluce a je to proces, ve kterém probíhá prohledávání prostoru doménových jmen. Způsob řešení dotazu může být rekurzivní (Obrázek 2) nebo iterativní (Obrázek 3). Oba způsoby se liší v principu, jak se server zachová v okamžiku, kdy nezná odpověď. Při rekurzivním řešení se server sám ptá dalších serverů na odpověď, kdežto při iterativním pouze vrátí adresy serverů, na které se má tazatel při řešení dotazu obrátit. Oba obrázky zobrazují posloupnost dotazů a odpovědí na jednotlivé servery během rezoluce.

Rezoluce začíná v cache paměti resolveru, pokud jí resolver disponuje. Pokud se dotaz nepodaří vyhledat v cache, resolver dotaz přepošle na nejbližší DNS server. Ten opět prohledá svou cache paměť (má-li ji) a pokud požadovaný záznam nalezne, posílá odpověď tazateli. Pokud záznam nenašel a jedná se o rekurzivní server, sám se dotáže kořenového serveru. Jedná-li se o iterativní server, pak tazateli odpoví, ať se obrátí na kořenový server. Postupným dotazováním se resolver dostane až k serveru, který má požadovaný záznam, v tu chvíli rezoluce končí a tazatel se dozvídá odpověď. Resolver i nejbližší DNS server, pokud se na rezoluci podílel, si odpověď ukládají do paměti cache.



Obrázek 2: Schéma rezoluce rekurzivního DNS dotazu



Obrázek 3: Schéma rezoluce iterativního DNS dotazu

Protokol

Největší část komunikace v systému DNS, klientské dotazy a odpovědi, se děje pomocí transportního protokolu UDP na portu 53. Protože se nejedná o spolehlivý protokol, je potřeba spolehlivost zajistit na úrovni aplikace – pomocí čísla v hlavičce paketu, a pokud se paket ztratí, poslat dotaz znovu.

Protokol DNS používá transportní protokol TCP na stejném portu pro aktualizaci zónových souborů mezi jednotlivými DNS servery.

2.2 DNS záznamy

DNS záznam (anglicky Resource Record, RR) je datová struktura, do které se ukládají doménová jména a další informace. Jednotlivé záznamy jsou ve formě textových souborů pro dané zóny uloženy na DNS serverech. Nejpoužívanější záznamy shrnuje v abecedním pořadí následující seznam:

- **A** (address) – mapuje doménové jméno na IPv4 adresu.
- **AAAA** (IPv6 address) – stejný jako záznam A, ale pro IPv6 adresu.
- **CNAME** (canonical name) – mapuje alias na kanonické jméno.
- **MX** (mail exchange) – určuje poštovní server pro danou doménu. Serverů může být více s různou prioritou.
- **NAPTR** (naming authority pointer) – mapuje řetězec na data. Používá se např. pro překlad telefonních čísel na SIP URI v IP telefonii.
- **NS** (name server) – určuje autoritativní DNS server v dané zóně. Serverů může být více, první bude primární, ostatní budou sekundární.
- **PTR** (pointer) – opak k záznamu A – mapuje IP adresu na doménové jméno, tzv. reverzní mapování.
- **SOA** (start of authority) – záznam, který musí mít každá zóna právě jeden. Obsahuje jméno primárního serveru a emailovou adresu správce zóny (zavináč je nahrazen tečkou).

Další důležité záznamy, které je třeba zmínit, jsou záznamy zajišťující zabezpečení systému DNS, například pomocí standardu DNSSEC. Mezi tyto záznamy patří DNSKEY, RRSIG, NSEC, NSEC3 a DS.

Každý záznam tvoří jméno, typ, třída, TTL a samotný obsah záznamu. TTL určuje dobu platnosti záznamu, přesněji řečeno jak dlouho mohou záznam uchovat cache servery. Po uplynutí této doby musí cache server kontaktovat autoritativní server a vyžádat si záznam znovu. TTL může obsahovat hodnotu 0, v tom případě záznam vůbec nesmí být uložen v cache (časté pro záznamy SOA). TTL hodnota se většinou neuvádí přímo pro konkrétní záznam, ale globálně pro celou zónu.

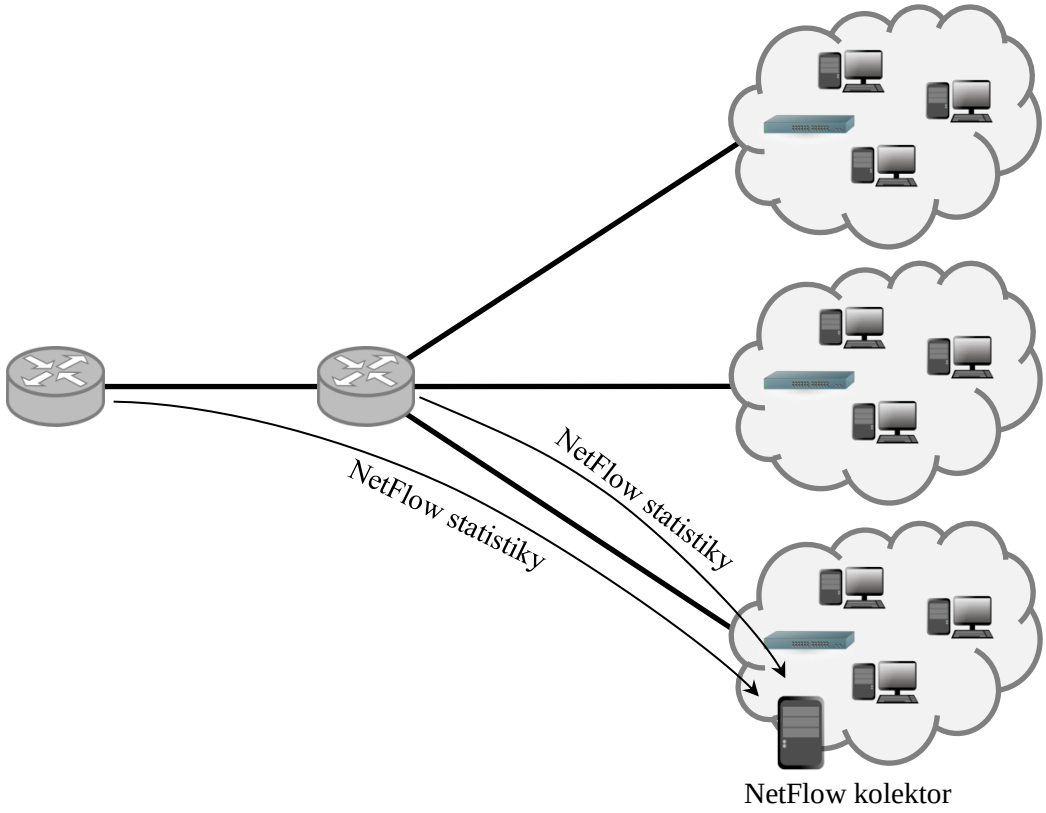
2.3 Zdroj síťových dat - NetFlow

NetFlow je protokol vytvořený společností Cisco Systems pro účely monitorování síťového provozu na základě tzv. síťových toků. Hlavními prvky technologie NetFlow jsou exportér, kolektor a komunikační protokol. Protokol při zpracovávání paketů využívá pouze hlaviček, samotný obsah paketů zahazuje. Díky tomu umožňuje administrátorům sítě v reálném čase zprostředkovat pohled na dění v jejich síti a jedná se tak o důležitou součást zabezpečení sítě.

Síťový tok je množina paketů, která se shoduje v následujících položkách:

- Zdrojová a cílová adresa,
- zdrojový a cílový port,
- logické rozhraní (tzv. ifIndex),
- typ protokolu na třetí vrstvě ISO/OSI modelu,
- hodnota ToS (Type of Service).

Z výše uvedeného vyplývá, že tok je jednosměrný mezi zdrojem a cílem. Pro jedno spojení typu klient-server tedy budou existovat dva toky.



Obrázek 4: NetFlow tradiční architektura

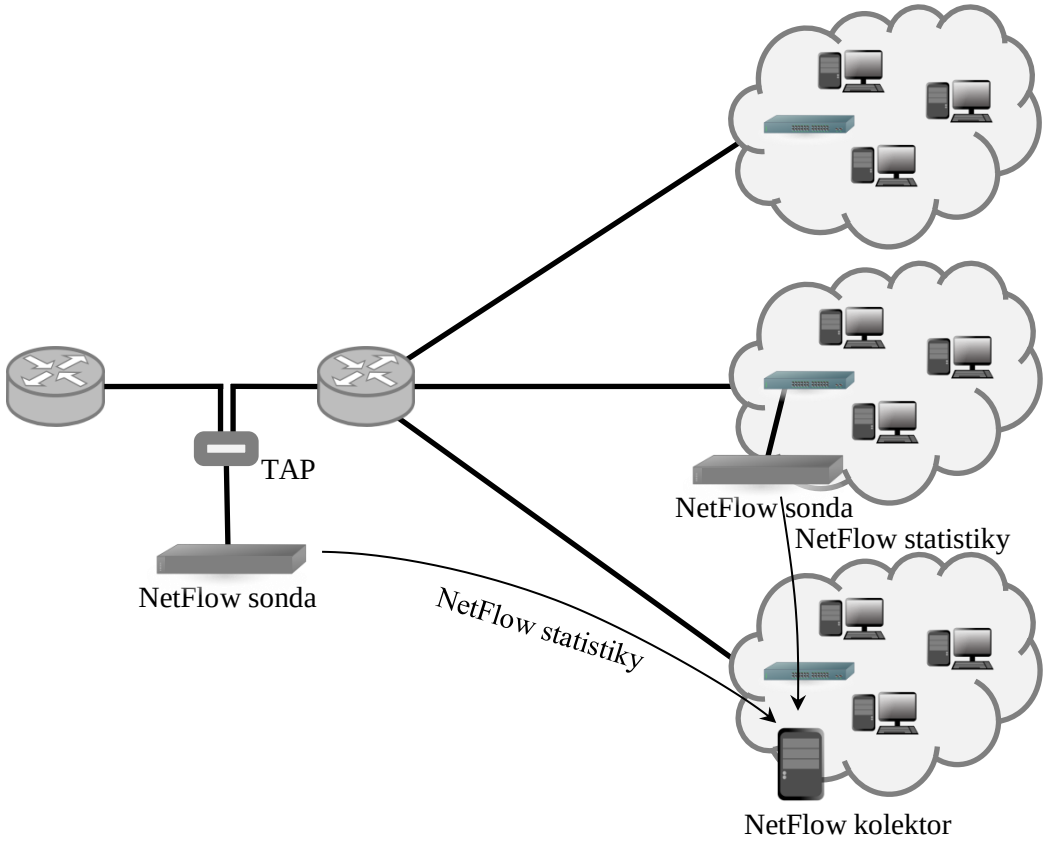
Exportér provádí samotné monitorování provozu. Vytváří záznamy o tocích nebo je aktualizuje v NetFlow cache. Tyto záznamy posílá kolektoru.

Exportér může být tvořen fyzickým síťovým zařízením nebo softwarově. Pokud se jedná o hardwarové zařízení, pak jde většinou o směrovače (viz Obrázek 4), které však musí být dostatečně výkonné a nejsou tedy nejlevnější (což odradí většinu menších a středních firem). Směrovače většinou provádí tzv. vzorkování dat, což znamená, že se pro výpočet statistik použije pouze zlomek původních dat. To může sice razantně snížit nároky na hardware, ale snižuje přesnost a pravděpodobnost odhalení bezpečnostního problému. Proto se používají také tzv. NetFlow sondy (viz Obrázek 5), což jsou specializovaná zařízení, která zajišťují funkci exportéru.

Kolektor sbírá záznamy od jednoho nebo více exportérů a stará se o jejich zpracování (případně i agregaci) a uložení ve formě statistik do své paměti. Nad těmito daty pak běží aplikace, která zařizuje vizualizaci těchto dat obvykle ve formě grafů či tabulek.

Komunikace mezi exportéry a kolektory probíhá pomocí protokolu NetFlow. Pro každý síťový tok je zaznamenáváno několik údajů, mezi nejdůležitější patří čas vzniku či ukončení toku a objem dat (počet paketů a bajtů).

Protokol NetFlow je vyvíjen spoustu let a za tu dobu bylo vytvořeno několik verzí. První masově používanou verzí byla verze 5 a podrobněji bude popsána v následující kapitole. Verze 6 přidala podporu pro tunelovaný provoz, verze 7 přibrala informace od switchů. Verze 9 zavedla větší flexibilitu pomocí šablon a dnes se hojně využívá. Všechny uvedené verze jsou proprietární. Verze 9 posloužila jako základ pro vznik nového otevřeného protokolu IPFIX (Internet Protocol Flow Information eXport).



Obrázek 5: NetFlow moderní architektura s využitím NetFlow sond

Škála činností, ke kterým lze technologii NetFlow využít, je široká, například pro

- monitorování sítí, aplikací a uživatelů,
- plánování sítí,
- bezpečnostní analýza,
- dlouhodobé ukládání informací o přenesených datech,
- účtování.

2.3.1 NetFlow packet v5

NetFlow packet se skládá z hlavičky a samotného záznamu. Údaje v obou částech se liší podle verze použitého protokolu. Podobu paketu verze 5 zobrazuje Tabulka 1, význam jednotlivých polí je následující:

- **version** – číslo verze formátu NetFlow
- **count** – počet exportovaných toků v tomto paketu
- **SysUptime** – aktuální doba provozu (v milisekundách) zařízení, jež data exportovalo
- **unix_secs** – aktuální počet sekund od 1. 1. 1970
- **unix_nsecs** – zbývající počet nanosekund od 1. 1. 1970
- **flow_sequence** – sekvenční čítač celkového počtu toků
- **engine type** – typ jednotky přepínající toky
- **engine id** – číslo slotu jednotky přepínající toky
- **sampling_interval** – první dva bity určují režim vzorkování, zbylých 14 obsahuje samotnou hodnotu intervalu vzorkování
- **srcaddr** – zdrojová IP adresa
- **dstaddr** – cílová IP adresa

- **nexthop** – IP adresa dalšího routeru po cestě (tzv. next hop router)
- **input** – SNMP index vstupního rozhraní
- **output** – SNMP index výstupního rozhraní
- **dPkts** – počet paketů v toku
- **dOctets** – celkový počet bytů síťové vrstvy v paketech toku
- **First** – doba provozu zařízení při vytvoření záznamu toku
- **Last** – doba provozu zařízení při přijetí posledního paketu toku
- **srcport** – zdrojový TCP/UDP port
- **dstport** – cílový TCP/UDP port
- **pad1** – nevyužito
- **tcp_flags** – TCP příznaky
- **prot** – identifikuje protokol vyšší vrstvy
- **tos** – typ služby
- **src_as** – číslo autonomního systému zdroje
- **dst_as** – číslo autonomního systému cíle
- **src_mask** – prefix zdrojové IP adresy
- **dst_mask** – prefix cílové IP adresy
- **pad2** – nevyužito

<table>
<tr><td colspan="6">bity</td></tr>
<tr><td>0 7</td><td>8 15</td><td>16 23</td><td>24 31</td><td></td><td></td></tr>
<tr><td colspan="6">hlavička</td></tr>
<tr><td>version</td><td>count</td><td colspan="4">SysUptime</td></tr>
<tr><td colspan="2">unix_secs</td><td colspan="4">unix_nsecs</td></tr>
<tr><td colspan="2">flow_sequence</td><td>engine type</td><td>engine id</td><td colspan="2">sampling_interval</td></tr>
<tr><td colspan="6">záznam</td></tr>
<tr><td colspan="2">srcaddr</td><td colspan="4">dstaddr</td></tr>
<tr><td colspan="2">nexthop</td><td colspan="2">input</td><td colspan="2">output</td></tr>
<tr><td colspan="2">dPkts</td><td colspan="4">dOctets</td></tr>
<tr><td colspan="2">First</td><td colspan="4">Last</td></tr>
<tr><td>srcport</td><td>dstport</td><td>pad1</td><td>tcp_flags</td><td>prot</td><td>tos</td></tr>
<tr><td>src_as</td><td>dst_as</td><td>src_mask</td><td>dst_mask</td><td colspan="2">pad2</td></tr>
</table>

Tabulka 1: Struktura paketu NetFlow v5

3 DNS anomálie

Pojmem DNS anomálie se označuje odchylka od standardního chování provozu na síti. Anomálie lze rozdělit na provozní, které mohou vzniknout vinou špatně nastaveného zařízení či jeho poruchou, a bezpečnostní, které se snaží zneužít chybu v zabezpečení systému. Mezi bezpečnostní DNS anomálie patří útoky typu DoS, DDoS či cache poisoning.

3.1 Metody detekce DNS anomálií

Metody používané pro detekci DNS anomálií využívají tzv. behaviorální analýzu sítě (Network Behavior Analysis, zkratka NBA). Základem NBA je vytvoření referenčního vzorku datových toků na síti. Tato data následně slouží k porovnání s provozem, který chceme analyzovat. Vzorky, které se statisticky odchylují od referenčního vzorku, budou prohlášeny za anomálie. Nevýhodou NBA je možnost falešné detekce. Dalším problémem může být samotný referenční vzorek, je třeba zajistit, aby neobsahoval žádné anomálie, jinak bude detekce neúspěšná (platí především pro učící se systémy).

Mezi metody behaviorální analýzy patří:

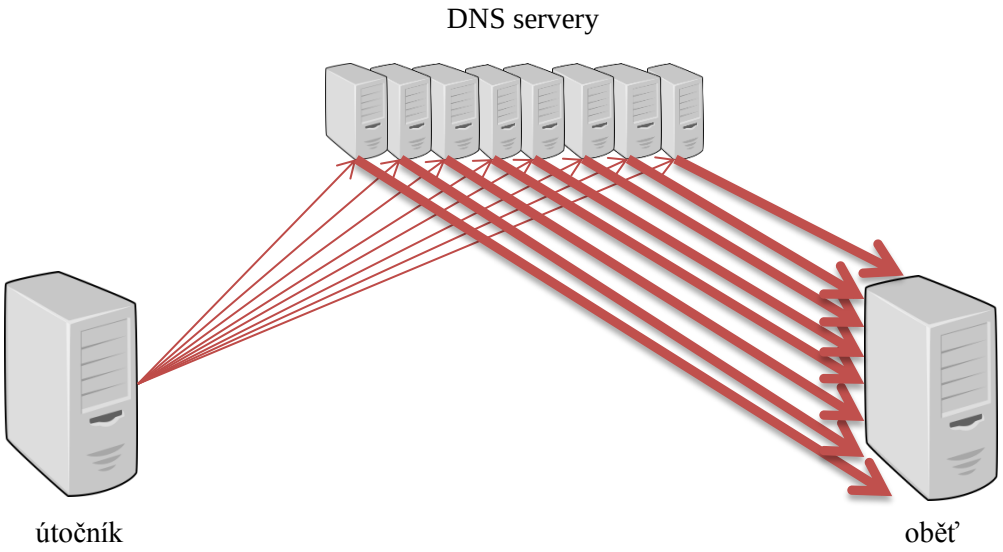
- Statistická metoda
- Metody srovnávání vzorů
- Metoda podobnosti
- Metoda entropie

3.2 Typy DNS anomálií

Denial of Service útok (zkratka DoS, česky odmítnutí služby) je typ útoku, při kterém se útočník snaží o zahlcení linky či serveru (vytížit procesor, paměť a podobně). Tím docílí situace, kdy server nestíhá zpracovat legitimní příchozí požadavky klientů, takže je musí zahazovat a server se tedy bude jevit jako nefunkční.

Distributed Denial of Service útok (zkratka DDoS, česky distribuované odmítnutí služby) je principiálně stejný jako útok DoS, ale liší se v rozložení zdrojových IP adres. U útoku DoS je spektrum zdrojových IP adres poměrně malé, kdežto u DDoS útoku přichází požadavky z hodně velkého rozsahu IP adres, většinou z velké části světa (za pomoci sítě napadených počítačů zvané botnet).

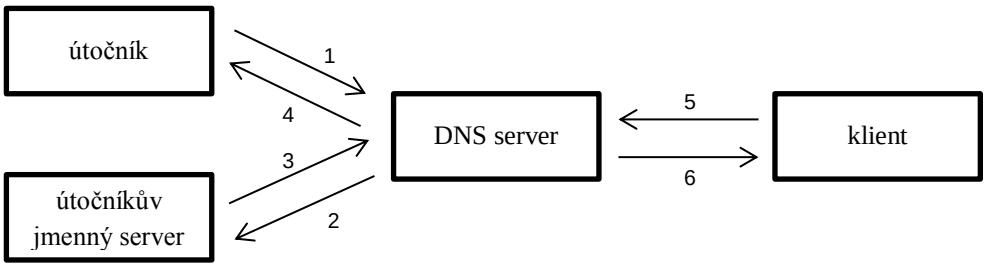
Útok DNS reflection (česky reflektivní DNS útok) je speciální případ DoS útoku. Jeho hlavní síla spočívá v tom, že je asymetrický, což v tomto případě znamená, že relativně malé množství infikovaných paketů dokáže vytvořit velmi silný útok. Útok využívá faktu, že odpověď na dotaz bývá větší než dotaz samotný. Teoreticky může být odpověď velká až 4 KiB. Útočník se pokusí zjistit, který nejkratší dotaz mu přinese největší odpověď a v ideálním případě mu může daný dotaz přinést až 100-násobnou odpověď. Zjednodušeně řečeno pak stačí v dotazu podvrhnout zdrojovou IP adresu za adresu cíle útoku, odeslat tento dotaz mnohokrát např. na veřejně přístupné DNS servery a ty se již postarají o zahlcení cílového serveru. Názorně tento útok zobrazuje Obrázek 6.



Obrázek 6: Schéma reflektivního DNS útoku

Útok cache poisoning spočívá v pokusu o podvržení falešné informace do vyrovnávací paměti DNS serveru. Jak ukazuje Obrázek 7, útočník vyšle takový dotaz (1) na server, který chce infikovat, aby jej daný server neuměl sám vyřešit (neměl ve vyrovnávací paměti) a při procesu rezoluce se musel obrátit na útočníkův server (2), který při odpovědi (3) využije např. pole RDATA v DNS paketu, a tak dostane podvrženou informaci. Klient, který se dotazuje napadeného serveru (5), pak dostane podvrženou informaci (6).

Útoku cache poisoning lze zabránit při využití DNSSEC – kontrola, zda byla odpověď podepsána tajným klíčem, což je možno ověřit veřejným klíčem (stojí za to zmínit, že v únoru 2014 bylo pomocí DNSSEC zabezpečeno pouze 36 % českých domén [1]).



Obrázek 7: Schéma útoku DNS cache poisoning

Přetížení DNS serveru je běžná anomálie zapříčiněná počtem DNS dotazů na server větším, než je server schopen obsloužit. Od útoku DoS se liší především rozložením zdrojových IP adres příchozích požadavků.

Chyba heartbleed (česky krvácející srdce) není DNS anomálie. V této práci je zmíněna a otestována z toho důvodu, že se objevila v průběhu testování nástroje a svým významem se jedná o jednu z nejzávažnějších chyb zabezpečení v historii internetu. Jde o zranitelnost v knihovně OpenSSL, objevená byla na přelomu března / dubna 2014. Chyba je označována jako nejzávažnější hned ze dvou důvodů: zaprvé tím, že zneužitím této chyby útočník může získat citlivé údaje včetně hesel či privátních klíčů k certifikátům a zadruhé svým neobvyklým rozsahem, údajně až na 17 % (v absolutních číslech cca půl milionu) všech zabezpečených serverů [2]. Dalším nepříjemným faktem je nejistota, zda k zneužití chyby u konkrétního serveru došlo nebo ne. Navíc tato chyba byla neodhalená řadu let (vznik kódu zodpovědného za tuto chybu se datuje na konec roku 2011 [3]). Jak

se ukázalo o měsíc později, chyba je stále neopravená na třistatisících serverech [4]. V této práci dojde k testu, zda lze uvedené metody použít na odhalení zneužití této chyby.

Princip zneužití je jednoduchý. V rámci zabezpečeného kanálu se přenáší tzv. TLS heartbeat, což jsou zprávy podobné ICMP echo zprávám sloužící mimo jiné k ověření spojení. TLS heartbeat požadavek obsahuje pole dat, které druhá strana přeposílá nezměněné zpět. Délka pole je variabilní, proto mu předchází dvoubajtová informace o jeho délce. Problém je ale v tom, že postižená knihovna nekontrolovala, zda zadaná délka není větší než délka celého paketu. Pokud tedy útočník v požadavku uvedl nejvyšší možné číslo, server mu v odpovědi odeslal vše, co v paměti RAM následovalo bezprostředně za tímto požadavkem. Útočník se tak mohl dozvědět téměř 64 KiB paměti serveru, kde mohly být uloženy libovolné citlivé údaje.

3.3 Detekce DNS anomálií pomocí podobnosti

Základní premisou této metody je časový harmonogram uživatelů Internetu, který je relativně fixní. To znamená, že počet paketů DNS provozu je den co den podobný. Pokud tedy vyneseme počty paketů DNS provozu do grafu a porovnáme křivky za každých 24 hodin, budou se lišit jen minimálně.

Jiná situace nastane ve chvíli, kdy dochází k nějakému útoku na systém DNS. Útok se projeví výrazným nárůstem DNS provozu, takže na základě porovnání křivky ve stejném časovém období lze rozhodnout, zda k útoku dochází, nebo ne. Naopak při přetížení DNS serveru k nárůstu nedojde a křivka se bude držet pod normálem.

3.3.1 Míra podobnosti

Klíčem k detekci výskytu DNS útoku je míra podobnosti, kterou představuje korelační koeficient ve statistikách. Ten porovnává dvě pole dat a vyhodnocuje, na kolik jsou si podobné. Výpočet se provede pomocí Rovnice 1,

$$\rho_{xy} = \frac{cov(X,Y)}{\sqrt{DX}\sqrt{DY}}$$

Rovnice 1: Výpočet Pearsonova korelačního koeficientu pomocí kovariance

kde *DX* a *DY* představují rozptyl pole dat *X* a *Y*. $cov(X,Y)$ reprezentuje kovarianci *X* a *Y*, která se vypočítá pomocí střední hodnoty *X* a *Y* (*E(X)* a *E(Y)*):

$$cov(X,Y) = E(XY) - E(X)E(Y)$$

Rovnice 2: Výpočet kovariance

Pearsonův korelační koeficient nabývá hodnoty v intervalu $\langle -1; 1 \rangle$. Hodnota -1 značí nepřímou závislost mezi veličinami, +1 naopak přímou závislost. Hodnota 0 znamená, že mezi veličinami není lineární závislost, což nevylučuje jiný způsob závislosti, který ale nelze vyjádřit lineární funkcí.

Předpokládáme lineární závislost mezi X a Y, musí tedy platit vztah $Y \approx aX + b$. Rovnice 1 lze přepsat do následujícího tvaru:

$$\rho_{xy} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\sqrt{E(Y^2) - E^2(Y)}}$$

Rovnice 3: Výpočet korelačního koeficientu

3.3.2 Korekce zachycených dat

Zachycená data mohou obsahovat určitou chybu, kterou může způsobit například selhání zařízení či napájení. Data, která jsou ovlivněna chybou, by mohla znehodnotit analýzu a vést ke špatným výsledkům. Bylo by tedy vhodné je odstranit.

První skupinu chybných dat tvoří body, které se již na prvních pohled liší od okolních (například představíme-li si tyto hodnoty vynesené do grafu). S korekcí těchto dat nám pomůže Rovnice 4.

Druhou skupinu tvoří tzv. nulové body. Jedná se o podobné body jako v předchozí skupině, ale jejich hodnota je nulová. Pro korekci těchto bodů poslouží Rovnice 5.

$$x_i = \begin{cases} \frac{1}{2p}\left(\sum_{j=0}^{i-1} x_j + \sum_{j=i+1}^{2p-i} x_j\right) \dots \text{pro } i \in [0,p) x_i > q\frac{1}{2p}\left(\sum_{j=0}^{i-1} x_j + \sum_{j=i+1}^{2p-i} x_j\right) \\ \frac{1}{2p}\sum_{j=1}^{m}\left(x_{i-j} + x_{i+j}\right) \dots \text{pro } i \in [p, n-p], x_i > q\frac{1}{2p}\sum_{j=1}^{m}\left(x_{i-j} + x_{i+j}\right) \\ \frac{1}{2p}\left(\sum_{j=2p-(n-i)}^{i-1} x_j + \sum_{j=i+1}^{n} x_j\right) \dots \text{pro } i \in (n-p, n], x_i > q\frac{1}{2p}\left(\sum_{j=2p-(n-i)}^{i-1} x_j + \sum_{j=i+1}^{n} x_j\right) \\ x_i \dots \text{jinak} \end{cases}$$

Rovnice 4: Korekce dat nenulové hodnoty

$$x_i = \begin{cases} \frac{1}{2p}\left(\sum_{j=0}^{i-1} x_j + \sum_{j=i+1}^{2p-i} x_j\right) \dots \text{pro } i \in [0,p) x_i = 0 \\ \frac{1}{2p}\sum_{j=1}^{m}\left(x_{i-j} + x_{i+j}\right) \dots \text{pro } i \in [p, n-p], x_i 0 \\ \frac{1}{2p}\left(\sum_{j=2p-(n-i)}^{i-1} x_j + \sum_{j=i+1}^{n} x_j\right) \dots \text{pro } i \in (n-p, n], x_i = 0 \\ x_i \dots \text{pro } x_i \neq 0 \end{cases}$$

Rovnice 5: Korekce dat nulové hodnoty

Obě rovnice předpokládají, že *X* je pole dat, *n* je délka posloupnosti dat. Parametry p a q volíme podle konkrétní situace.

3.3.3 Způsob porovnání podobnosti

Princip metody je založen na porovnání podobnosti dvou vzorků dat. Měli bychom tedy mít referenční vzorek dat. Tento vzorek by mělo tvořit *r* polí hodnot zaznamenaných v průběhu dne, které budeme porovnávat. Počet polí hodnot by neměl být ani velký, ani malý. Příliš velké *r* zbytečně sníží efektivitu, naopak při příliš malém *r* nebudou výsledky dostatečně přesné.

Vzhledem k tomu, že ne všechny dny v roce jsou stejné, dá se předpokládat, že s DNS provozem to bude podobné. Proto by bylo vhodné mít několik referenčních vzorků pro eliminaci tohoto kritéria. Za referenční hodnoty by se tedy daly považovat tři skupiny vzorků zachycených v pracovní dny, o víkendech a svátcích. Každé skupině vzorků je třeba určit minimální práh podobnosti.

Samotné porovnání proběhne jednoduše spočítáním míry podobnosti v každém časovém úseku z porovnávaných hodnot a jim odpovídajícím hodnotám referenčním. Pro každý vzorek určíme, zda došlo k překročení minimálního prahu podobnosti. Pokud alespoň pro jeden vzorek k překročení prahu podobnosti nedošlo (vzorek se shoduje alespoň s jedním referenčním vzorkem), lze prohlásit, že DNS útok nenastal.

3.4 Detekce DNS anomálií pomocí entropie

Datový tok, ve kterém chceme detekovat anomálie, budeme reprezentovat několika rozměrným vektorem, kde každý rozměr bude reprezentovat jednu vlastnost datového toku (například zdrojová IP adresa, port). Pro každou vlastnost, označovanou jako Feature Property [5], můžeme spočítat hodnotu entropie[1]. Při běžném provozu v datovém toku se bude entropie držet v určitém rozpětí. Naopak, pokud dochází k nějakému typu DNS útoku, entropie se změní.

Změnu entropie zapříčiní jiné složení paketů v datovém toku. Při útoku typu DoS dojde ke změně entropie zdrojové a cílové IP adresy. Útok cache poisoning se podepíše na změně entropie cílové IP adresy a portu. [3]

3.4.1 Výpočet entropie

Mějme prvky $X_1, X_2, \dots, X_n$. Jejich pravděpodobnost výskytu je $P_1, P_2, \dots, P_n$ při splnění podmínek $0 < P_i \leq 1$ pro $i = 1,2, \dots, n$ a $\sum_{i=1}^{n} p_i = 1$. Entropii pak lze spočítat pomocí vztahu:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Rovnice 6: Výpočet entropie

[1] Entropie = míra neuspořádanosti

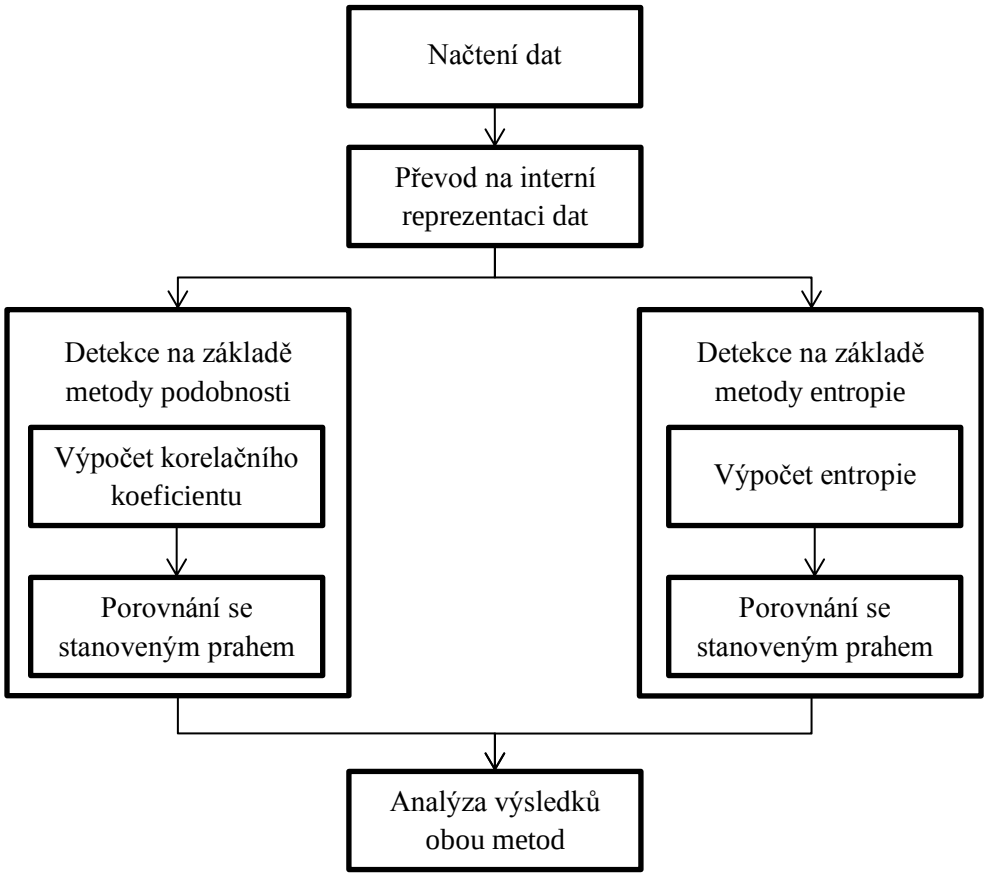
4 Návrh nástroje

Cílem této práce je navrhnout a implementovat aplikaci, která na základě referenčního vzorku dat bude v testovaném vzorku hledat DNS anomálie na základě metody podobnosti a entropie. Vzhledem k povaze úkolu se bude jednat o konzolovou aplikaci.

Aplikace se bude skládat z několika částí, které se budou starat o určité sekce analyzátoru. Z obecnějšího pohledu se bude jednat o 4 větší celky, konkrétně půjde o

- načtení a interpretaci vstupních dat,
- analýzu pomocí metody podobnosti,
- analýzu pomocí metody entropie a
- vyhodnocení výsledků obou metod.

První část aplikace se bude muset postarat o korektní načtení vstupních dat, která mohou být ve formátech NetFlow nebo pcap. Z toho důvodu je bude zřejmě nutné převést na určitou (společnou) formu reprezentace, se kterou budou schopny pracovat části implementující obě metody detekce.



Obrázek 8: Princip činnosti analyzátoru

Načtená data dostanou k analýze na sobě nezávislé jednotky implementující metodu podobnosti a metodu entropie. Ty budou mít v principu velmi podobnou funkci: analyzovat předaná data a na základě stanovených konstant rozhodnout, zda nastala anomálie.

Metoda podobnosti tohoto cíle dokáže, když spočítá v daném intervalu z počtů paketů korelační koeficient. Ten posléze porovná se stanoveným minimálním prahem podobnosti a bude-li vyšší nebo roven, tak anomálie nenastala. Analýza je tedy možná postupně interval po intervalu.

Metoda entropie na to půjde trochu odlišně. Pro výpočet entropie ji stačí pouze analyzovaná data z konkrétního intervalu. K tomu ještě potřebuje znát minimální a maximální entropii, aby mohla rozhodnout, za je právě spočítaná entropie v rozpětí daném minimální a maximální entropií. Nejprve tedy analyzuje celá referenční data a to tak, že pro každý interval spočítá entropii. Ze všech intervalů pak zjistí minimální a maximální entropii a tu si uloží. A přesně tyto dvě hodnoty použije, když začne analyzovat testovací data, aby rozhodla o výskytu anomálie.

V poslední části aplikace bude zpracován výsledek obou metod, který bude rozumnou formou prezentován uživateli. Zřejmě bude také vhodné mít možnost uložit detailní výsledky analýzy do souboru, což znamená uložit jednotlivé dílčí výsledky korelačního koeficientu a entropie analyzovaného a referenčního souboru.

Funkčnost aplikace a chování jednotlivých metod bude otestováno na NetFlow datech zachycujících zesilující DNS útok s daty ze stejné linky před nebo po útoku. Obě metody také budou otestovány na útoku nesouvisejícím s DNS – zneužití chyby označované jako Heartbleed a bude tak ověřeno, zda jsou metody účinné pro detekci této zranitelnosti.

5 Implementace

Implementace aplikace vychází z předchozí kapitoly. První důležitou volbou bylo zvolit vhodný programovací jazyk. Rozhodnutí vycházelo z nutnosti interpretovat vstupní data ve formátech PCAP a NetFlow. Pro oba formáty existují hotové knihovny pro jazyk C/C++, takže volba padla na něj. Nejprve bylo nutné se důkladně seznámit s oběma knihovnami, jejich popis je v následujících dvou kapitolách.

5.1 pcap API

Formát pcap (**p**acket **cap**ture) se používá k zachycení síťového provozu ve formě celých paketů. S formátem pcap lze pracovat pomocí pcap API (aplikačně programové rozhraní), které poskytuje nástroje k zachycení a filtrování paketů. Na unixových systémech toto rozhraní implementuje knihovna libpcap, na Windows existuje WinPcap. S tímto API pracuje mnoho aplikací, např. tcpdump, Wireshark, Microsoft Network Monitor, nmap, Kismet a další.

Soubory uložené v tomto formátu (tzv. dump) mají nejčastěji příponu .pcap a MIME typ application/vnd.tcpdump.pcap.

pcap API se skládá z definic struktur, konstant, datových typů a funkcí. Nejdůležitějším částem API se budou věnovat následující odstavce.

Struktury

pcap_file_header obsahuje hlavičku souboru se zachycenými daty. Kromě čísla verze použité knihovny libpcap obsahuje také například informaci o časové zóně, kvůli korekci časových údajů či maximální délku části uloženého paketu.

pcap_pkthdr ukládá metadata zachyceného paketu v souboru. Konkrétně se jedná o čas zachycení paketu (ve formě unixového času – počet sekund od 1. 1. 1970, tzv. timestamp ve struktuře timeval), dále pak velikost zachyceného paketu a datagramu.

pcap_stat uchovává statistické hodnoty rozhraní – počet přijatých / zahozených paketů.

pcap_if je jednosměrně vázaný seznam rozhraní, který o každém rozhraní ukládá název, popis, adresu (struktura pcap_addr) a příznaky.

pcap_addr je jednosměrně vázaný seznam adres rozhraní. Adresa se skládá ze struktur sockaddr, které ukládají síťovou adresu, masku, adresu všesměrového vysílání (tzv. broadcast) a cílovou adresu.

Datové typy

pcap_t je deskriptor otevřené instance zachytávání.

pcap_dumper_t je deskriptor souboru pro uložení/načtení zachycených dat

pcap_if_t je položka v seznamu rozhraní.

pcap_addr_t je položka v seznamu adres rozhraní.

Funkce

Funkce **pcap_open_offline** slouží k otevření uloženého souboru ve formátu pcap. Vrací ukazatel *pcap_t* na otevřený soubor.

pcap_next vrátí ukazatel na řetězec dat dalšího dostupného paketu. Přes argument jsou dostupná metadata paketu.

pcap_dump_open otevře soubor pro uložení paketů do souboru.

pcap_dump uloží paket do souboru.

5.2 nfreader

Knihovna nfreader slouží pro čtení uložených NetFlow záznamů z kolektoru. Vychází z nástroje nfdump, který podporuje NetFlow verze 5, 7 a 9. Knihovna obsahuje několik struktur, datových typů a funkcí pro práci s NetFlow záznamem.

Datové typy a struktury

nf_file_t je struktura pro otevřený dump soubor.

Struktura **ip_addr_s** (resp. datový typ **ip_addr_t**) ukládá IP adresu (v4 i v6).

Struktura **master_record_s** (resp. **master_record_t)** obsahuje kompletní data o NetFlow záznamu.

Struktura **stat_record_s** (resp. **stat_record_t**) sdružuje statistické informace o tocích (počet toků, bytů, paketů, atp.).

Funkce

nf_open otevře zadaný nfdump soubor nebo standardní vstup pro čtení NetFlow toků.

nf_close zavře zadaný nfdump soubor.

nf_next_record načte další záznam z otevřeného souboru do struktury master_record_t.

nf_get_stats získá statistiky ze zadaného nfdump souboru.

5.3 Čtení dat

Pro načtení vstupní dat (ať už analyzovaných či referenčních / historických) byla využita knihovna libpcap (WinPcap) nebo nfreader podle toho, zda se jedná o pcap dump soubor nebo NetFlow dump soubor. V obou případech se nejprve provede otevření souboru a následně zpracování jednotlivých paketů nebo toků.

Knihovna libpcap (WinPcap) zpracovává jednotlivé pakety ze vstupního souboru a samotný paket vrací ve formě ukazatelu na řetězec surových data, která nejsou nijak zpracovaná. Data jsou tedy uložena v binární podobě, jak je definuje síťový model ISO/OSI počínaje linkovou vrstvou, resp. model TCP/IP od vrstvy síťového rozhraní (dnes prakticky Ethernet). Jedná se tedy o rámec (ochuzený o preambuli a SFD), který v sobě zapouzdřuje IP paket, atd. Z tohoto rámce je pro nás zajímavé pouze dvoubajtové pole typu vyššího protokolu (musí obsahovat identifikátor IPv4 nebo IPv6 protokolu).

V IP paketu jsou z pohledu tohoto nástroje mnohem zajímavější informace. Pro jejich interpretaci lze použít strukturu ip (definovanou v knihovně netinet/ip.h). Konkrétně nás zajímá zdrojová a cílová IP adresa a velikost paketu. Ovšem pouze u paketů, které zapouzdřují UDP datagram (DNS požadavek je zapouzdřen v UDP datagramu). V něm jsou zajímavé informace pouze zdrojový a cílový port. Abychom mohli datagram použít, musí být jeden z portů 53 (standardní port služby DNS pro dotazy). Pokud jsou zmíněné podmínky splněny, jedná se tedy o IPv4 nebo IPv6 DNS paket a přidáme další záznam do pole struktur packet_headers.

Struktura packet_headers tvoří interní reprezentaci dat, přesněji řečeno jednoho záznamu, z načteného souboru. Struktura obsahuje pole nutná pro výpočet korelačního koeficientu či entropie. Konkrétně by to tedy měla být časová značka paketu / toku, velikost paketu, zdrojová a cílová IP adresa a zdrojový a cílový port. Ačkoli metoda podobnosti vyžaduje prakticky pouze 2 prvky z této struktury, nemělo by smysl vytvářet reprezentaci pro každou metodu zvlášť a provádět tak duplikaci dat (či ukazatelů).

Knihovna nfreader čte jednotlivé toky ze vstupního souboru a vrací je ve formě struktury master_record_t. Je potřeba pouze zkontrolovat, zda se jedná o IPv4/IPv6 DNS tok, podobně jako u pcap souboru, a pokud ano, jednoduše překopírovat data ze struktury master_record_t a vložit je do pole struktur packet_headers.

5.4 Metoda podobnosti

Metoda podobnosti využívá k detekci DNS anomálií počty paketů v jednotlivých intervalech. Tyto počty porovnává v historických a analyzovaných datech pomocí tzv. Pearsonova korelačního koeficientu. Nejprve je třeba projít obě pole dat a zjistit počty paketů v jednotlivých intervalech. V každém intervalu z počtu paketů spočítáme střední hodnotu, rozptyl a směrodatnou odchylku, z čehož spočítáme kovarianci. Z těchto údajů můžeme podle Rovnice 1 vypočítat Pearsonův korelační koeficient. Koeficient pak porovnáme se stanovaným prahem, který by měl překročit. Pokud se tak nestalo, zřejmě došlo k DNS anomálii.

5.5 Metoda entropie

Detekce DNS anomálií na základě metody entropie nejprve vyžaduje z referenčních dat spočítat minimální a maximální entropii. Při analyzování testovacích dat pak dochází k výpočtu entropie v daném intervalu. Spočítaná entropie musí být v rozmezí minimální a maximální entropie, v opačném případě zřejmě došlo k nějaké DNS anomálii.

Pro výpočet entropie se použije Rovnice 6. Entropie bude spočítána pro každou složku (zdrojová a cílová IP adresa a port, velikost paketu) zvlášť a celková entropie bude získána váženým součtem dílčích entropií. Pro každou složku je tedy třeba spočítat pravděpodobnost výskytu. Zjistíme unikátní záznamy v každé složce a kolikrát se v dané složce daný záznam vyskytl. Pravděpodobnost výskytu daného záznamu je pak počet výskytů děleno celkovým počtem záznamů.

5.6 Výstup aplikace

Jak již bylo zmíněno, jedná se o konzolovou aplikaci, takže nemá žádný grafický výstup. Základní zjištěné informace vypisuje na standardní výstup ve formátu:

```
[metoda] <počáteční interval;konečný interval>: Příčina anomálie
```

Jakmile aplikace dokončí analyzování referenčního souboru, vypíše také spočítanou minimální a maximální entropii, aby bylo jasné, když se zjistí anomálie a její příčinou je nízká či vysoká entropie, nakolik se tato odlišuje od minima nebo maxima.

Tento formát by byl ale nedostatečný pro vytvoření grafu entropie nebo míry podobnosti. Aplikace je proto schopna vytvořit několik souborů dat (pro průběh entropie referenčního i analyzovaného souboru a také pro hodnoty podobnosti), z nichž je možné tyto grafy sestrojit. Tyto soubory se negenerují automaticky, je potřeba je explicitně povolit speciálním parametrem příkazové řádky, viz Uživatelská příručka. Vytvořené soubory jsou ve formátu CSV a je tak snadné s nimi pracovat v jiných aplikacích (např. Microsoft Excel).

Až se dokončí analýza, aplikace informuje uživatele o celkových počtech referenčních a analyzovaných paketů. Obrázek 9 ukazuje možný výstup aplikace.

Aplikace analyzuje jednu sadu dat – referenční a analyzovaný soubor. V případě NetFlow dat bývají data rozdělená do několika souborů např. po 5 minutovém období. Ruční spouštění aplikace pro každé období by bylo neefektivní a náchylné k chybám, proto součástí aplikace vznikl skript,

který zařídí automatickou analýzu všech souborů v dané složce s referenčními i analyzovaným soubory. Ovládání tohoto skriptu je popsáno taktéž v příloze v kapitole Uživatelská příručka.

```
$ ./dnsad -r test2.pcap -f test3.pcap
Minimum entropy: 0,017617
Maximum entropy: 1,021375
[similarity] <1394039590:1394039620>: Similarity is -0.258016
[similarity] <1394039620:1394039650>: Similarity is 0.269229
[similarity] <1394039650:1394039680>: Similarity is -0.213579
[similarity] <1394039680:1394039710>: Similarity is -0.0795668
[similarity] <1394039710:1394039740>: Similarity is -0.0391955
[similarity] <1394039740:1394039770>: Similarity is -0.188114
[similarity] <1394039770:1394039800>: Similarity is -0.133846
[similarity] <1394039800:1394039830>: Similarity is 0.33954
[similarity] <1394039830:1394039860>: Similarity is -0.379954
[similarity] <1394039920:1394039950>: Similarity is 0.0525091
[similarity] <1394039950:1394039980>: Similarity is 0.0892281
[similarity] <1394039980:1394040010>: Similarity is 0.288353
[similarity] <1394040010:1394040040>: Similarity is -0.476508
[similarity] <1394040040:1394040070>: Similarity is -0.128564
[similarity] <1394040070:1394040100>: Similarity is -0.989743
Number of reference DNS packets:            1155
Number of analyzed DNS packets:             1237
```

Obrázek 9: Ukázka možného výstupu aplikace

6 Testování

Tato kapitola popisuje způsob testování, na jakých datech probíhalo a s jakými výsledky. Nejprve proběhly testy na sadách dat pro reflektivní DNS útok, navíc bylo otestováno zneužití chyby HeartBleed. Zdrojem dat ve formě NetFlow záznamů byly exportéry v počítačové síti FIT VUT v Brně, data byla před spuštěním aplikace anonymizována.

Všechny výpočty probíhaly po půlminutových intervalech. Pro výpočet entropie byly stanoveny následující složky: zdrojová IP adresa, cílová IP adresa, zdrojový port, cílový port, velikost paketu. První čtyři složky jsou nejdůležitější, takže se na celkové entropii podepisují každá 22 %, pro velikost pak zbývá 12 %. Práh podobnosti byl stanoven na 0,35, což je hodnota, při které jsou si data alespoň trochu podobná. Nižší už o moc být nemůže, protože při hodnotě 0 si data podobná nejsou vůbec (není mezi nimi lineární závislost). Pokud by byla vyšší, aplikace by vyžadovala striktnější podobnost dat a zřejmě by častěji detekovala anomálie, přičemž by mohlo jít o falešné poplachy.

Soubory s daty byly rozděleny po pětiminutových intervalech, výsledné grafy vznikly spojením dílčích intervalů z CSV souborů vygenerovaných aplikací. Některé soubory ale nekončili přesně s daným intervalem, proto grafy nejsou násobkem pěti minut.

6.1 Reflektivní DNS útok

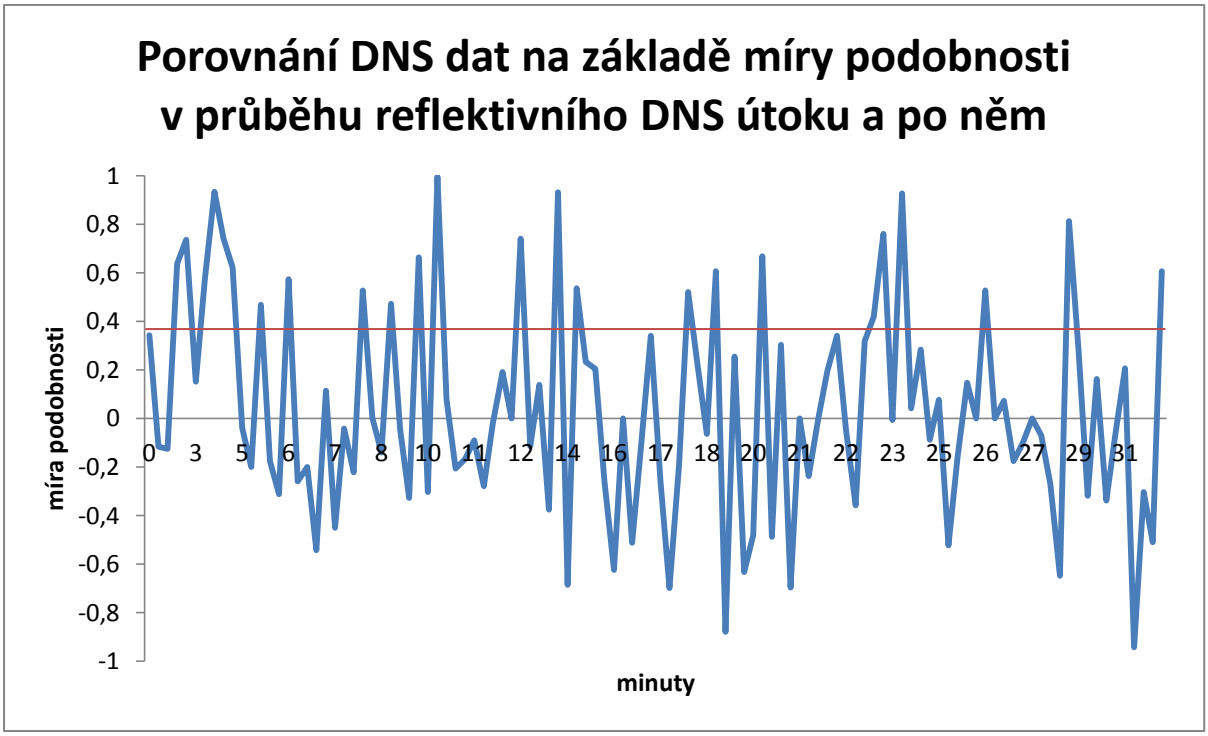
Tento typ útoku byl otestován na dvou sadách dat, referenční a analyzovaná data pochází ze stejné linky v rozpětí několika desítek minut.

6.1.1 První sada dat

Referenční data pro první test obsahují 60 minut IP toků a pochází z 8. 10. 2013 v době po útoku, který je zaznamenaný v analyzovaných datech, které mají trvání 30 minut.

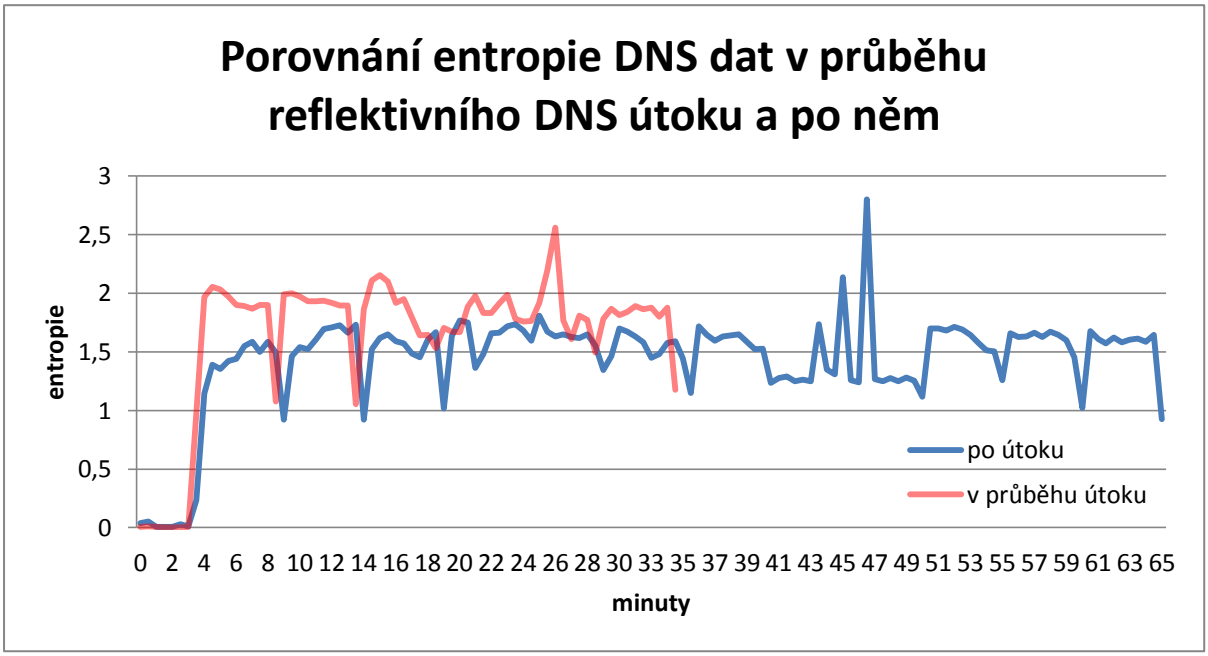
Vzhledem ke kratší době analyzovaných dat je míra podobnosti vypočítána pouze po dobu, kdy lze srovnávat obě pole dat, tedy 30 minut. Entropii lze spočítat pro každé pole dat samostatně, takže graf pro metodu entropie bude vykreslen pro celých 60 minut referenčních dat.

Obrázek 10 zobrazuje vývoj míry podobnosti DNS dat před a po útoku včetně prahu podobnosti. Na základě tohoto grafu se dá usoudit, že si data nejsou podobná, protože korelační koeficient překročí stanovený práh pouze v 21,8 % vzorků. Ve zbylých 78,2 % vzorků je pod prahem, což naznačuje pravděpodobný výskyt anomálie. Aritmetický průměr vzorků je 0,0381.



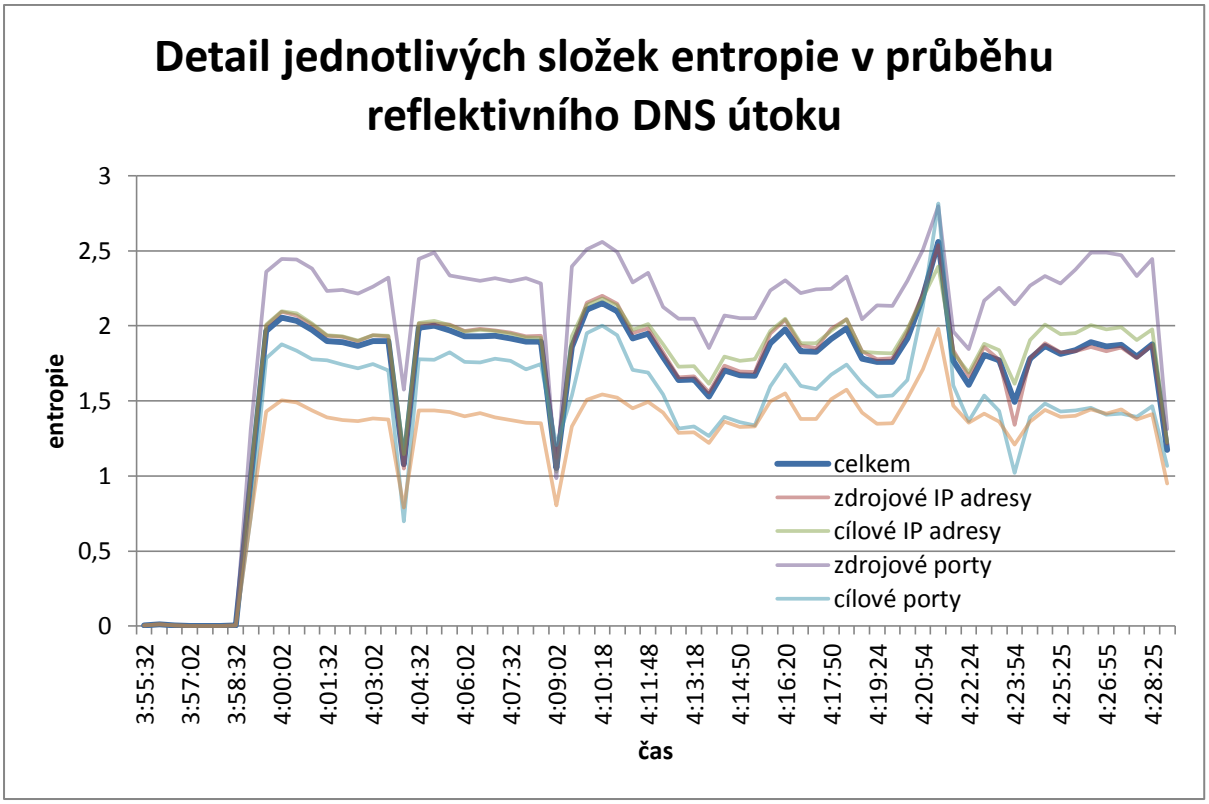
Obrázek 10: Porovnání DNS dat na základě míry podobnosti v průběhu reflektivního DNS útoku a po něm (test 1)

Metoda entropie už na první pohled potvrdila výsledek metody podobnosti. Obrázek 11 ukazuje, že entropie v průběhu útoku byla, kromě prvních pěti minut, celou dobu vyšší než entropie po útoku. V prvních třech minutách bylo v analyzovaných i referenčních datech zachyceno pouze minimum toků. Další minuty dokazují výskyt anomálie.

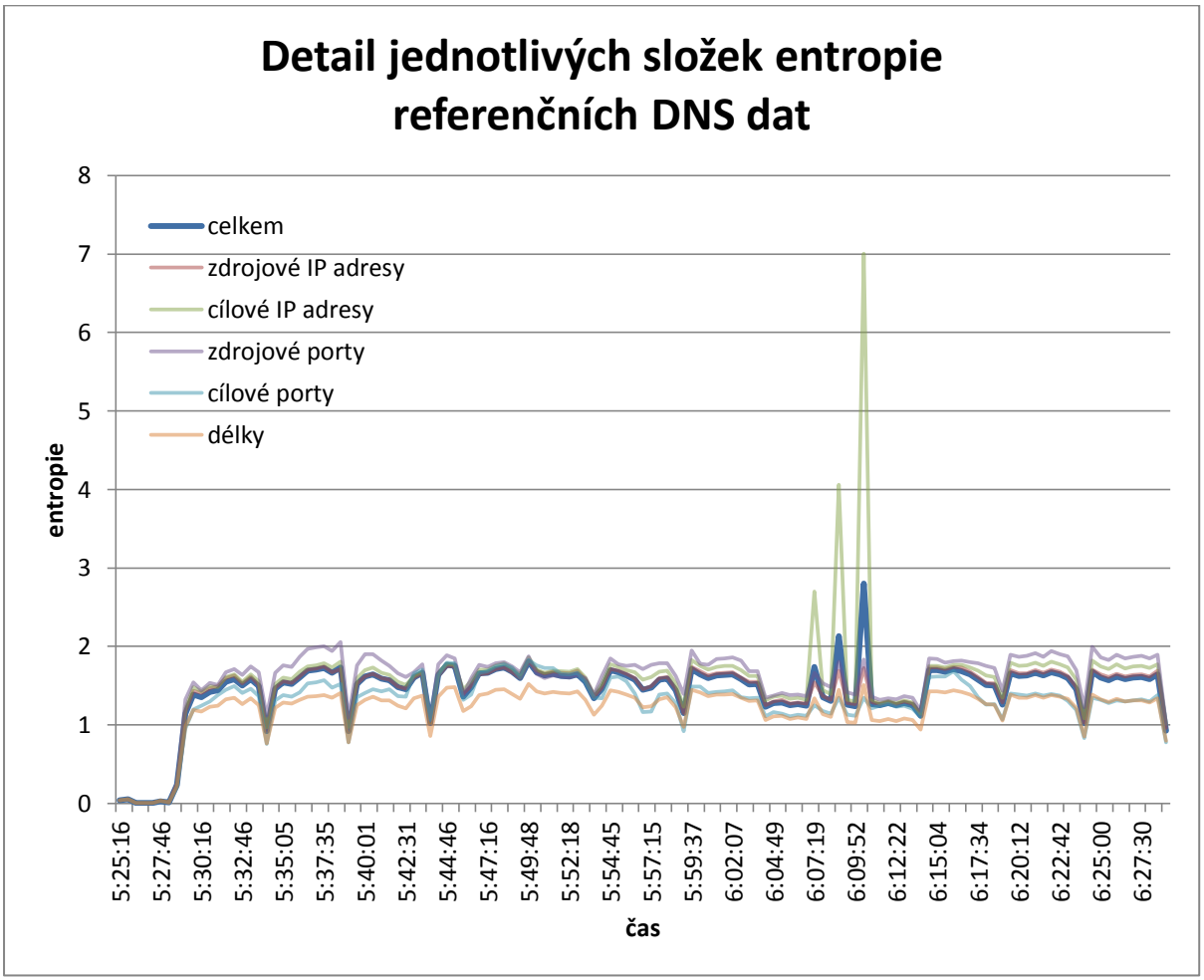


Obrázek 11: Porovnání entropie DNS dat v průběhu reflektivního DNS útoku a po něm (test 1)

Bohužel při podrobnějším zkoumání jednotlivých složek entropie není znatelná žádná charakteristická změna některé ze složek, viz Obrázek 12. Naopak při pohledu do chování entropie čistých dat okolo 42. až 48. minuty lze pozorovat nějakou anomálii. Detailní pohled na jednotlivé složky odhaluje nárůst entropie cílových IP adres (Obrázek 13).



Obrázek 12: Detail jednotlivých složek entropie v průběhu reflektivního DNS útoku (test 1)



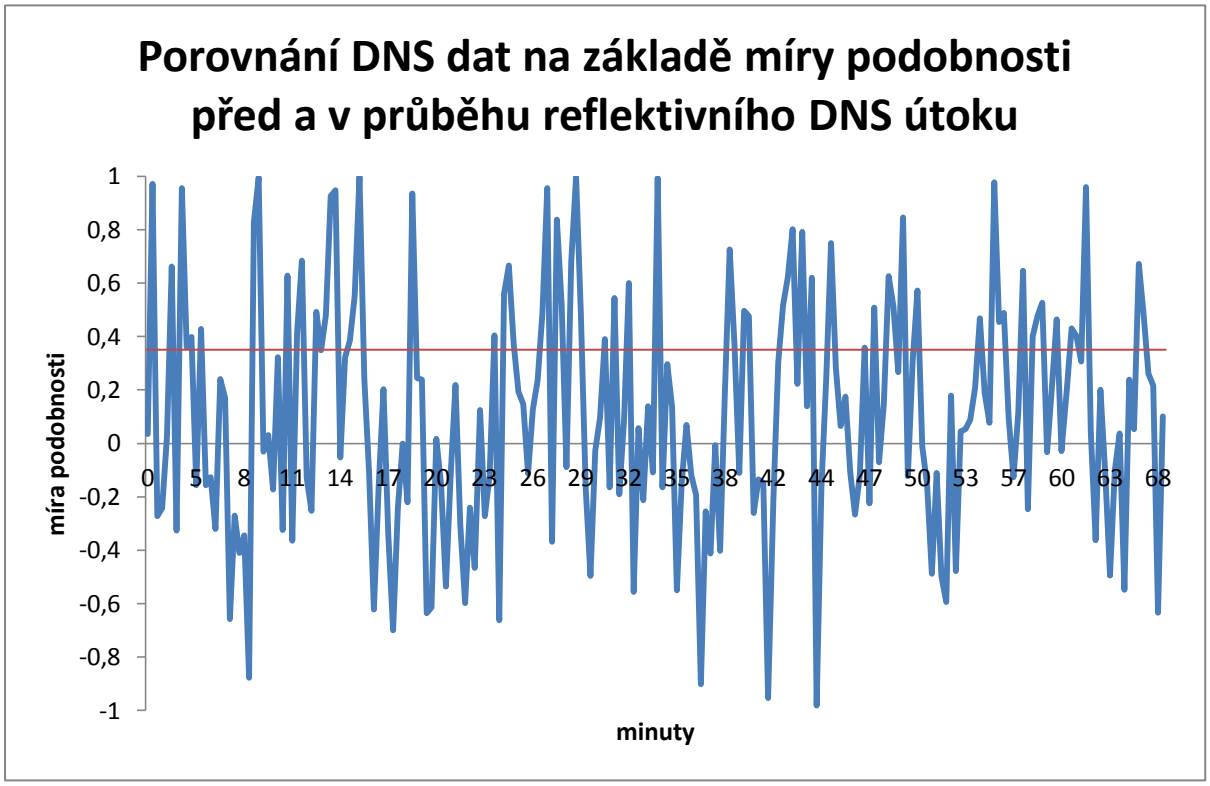
Obrázek 13: Detail jednotlivých složek entropie referenčních DNS dat (test 1)

Porovnáním výsledků obou metod docházím k závěru, že se útok vyskytl po celou dobu trvání analyzovaných dat.

6.1.2 Druhá sada dat

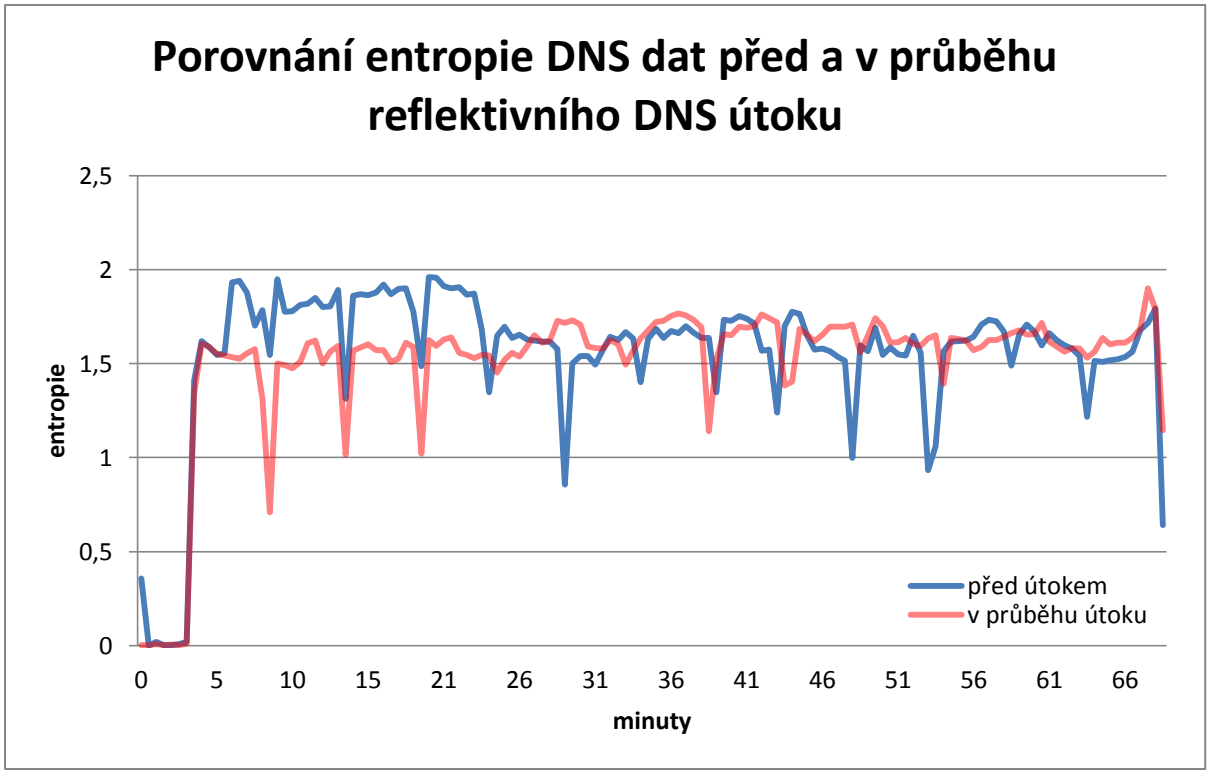
Druhá sada dat obsahuje 13x5 minut, tedy 65 minut referenčních dat ze dne 7. 10. 2013 zachycených těsně před útokem, a 65 minut dat analyzovaných na DNS anomálie.

Opět se nejprve zaměříme na metodu podobnosti, která v tomto případě měla dostatek dat pro analýzu celého vzorku testovaných dat. Obrázek 14 ukazuje, že je míra podobnosti často pod prahem, což by mohlo značit výskyt anomálie. Ze statistického pohledu vzato je 30,2 % vzorků nad prahem podobnosti, aritmetický průměr je nepatrně vyšší než u prvního testu, konkrétně 0,1062.



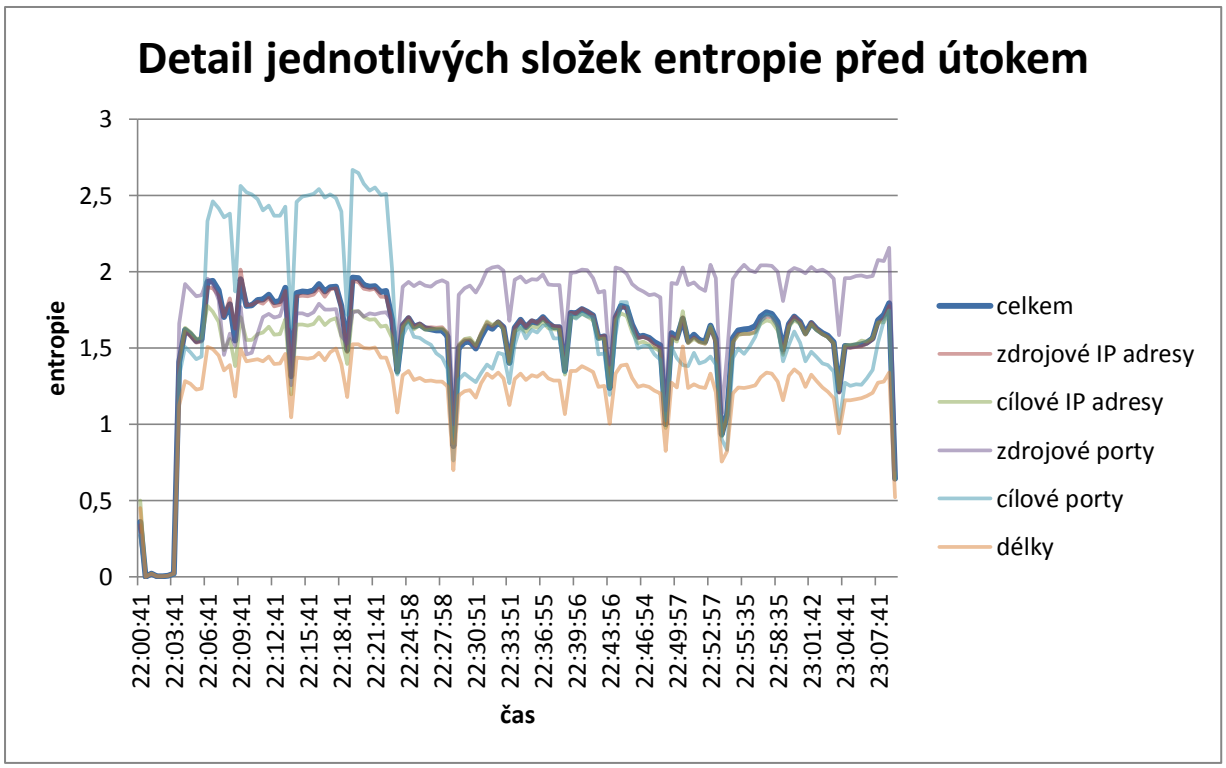
Obrázek 14: Porovnání DNS dat na základě míry podobnosti před a v průběhu reflektivního DNS útoku (test 2)

Na základě spočítané entropie (viz Obrázek 15) lze usoudit, že anomálie nastala mezi 5. a 26. minutou. V tomto období je entropie z analyzovaného souboru (v průběhu útoku) pod úrovní entropie z referenčního souboru (před útokem). V dalších minutách se entropie drží v mezích daných referenčním souborem, takže v té době útok zřejmě už neprobíhal.

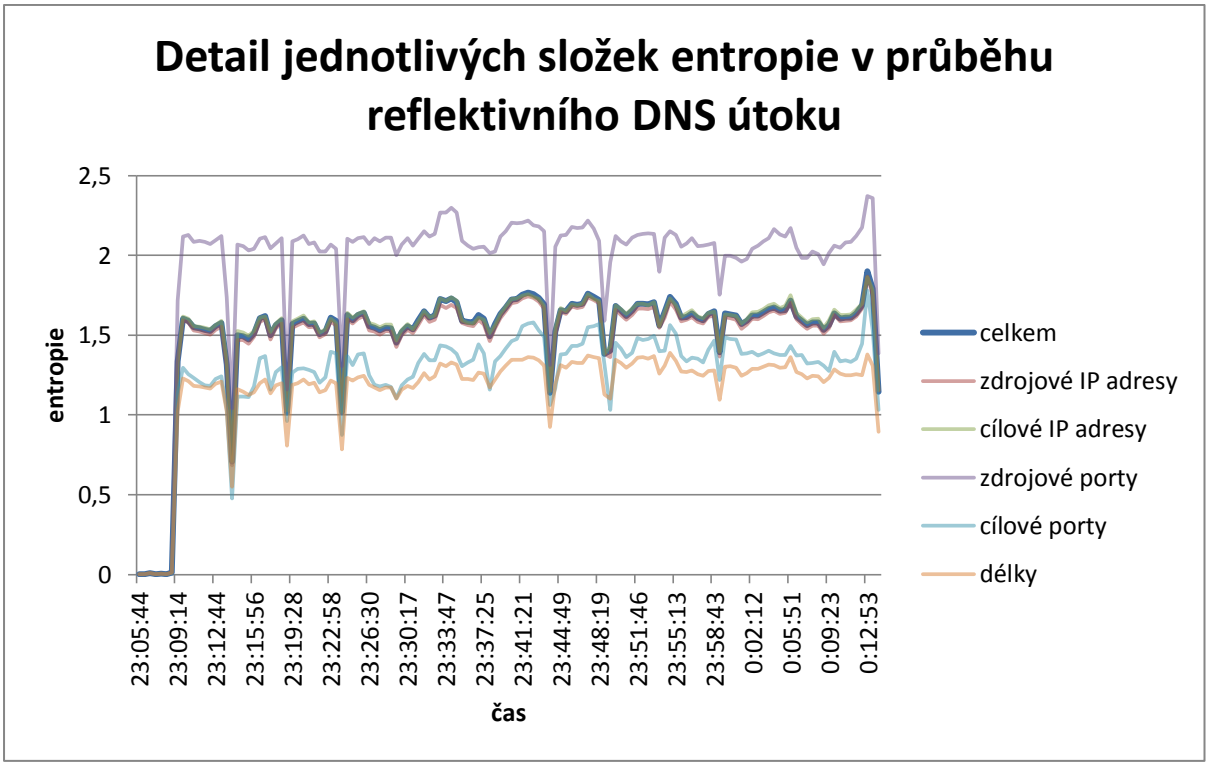


Obrázek 15: Porovnání entropie DNS dat před a v průběhu reflektivního DNS útoku (test 2)

Jednotlivé složky entropie referenčních dat ukazuje Obrázek 16. Mělo by se jednat o čistá data, ovšem od 6. do 23. minuty vidíme výrazný nárůst entropie cílových portů, ačkoliv to není jediná složka, kde je vidět nárůst. Naneštěstí je to prakticky v identickém čase, kdy jsme detekovali anomálii v analyzovaném souboru. Vyvstává tedy otázka, zda v analyzovaném souboru k anomálii skutečně došlo, nebo byla detekována vlivem nesprávných referenčních dat.



Obrázek 16: Detail jednotlivých složek entropie před útokem (test 2)



Obrázek 17: Detail jednotlivých složek entropie v průběhu reflektivního DNS útoku (test 2)

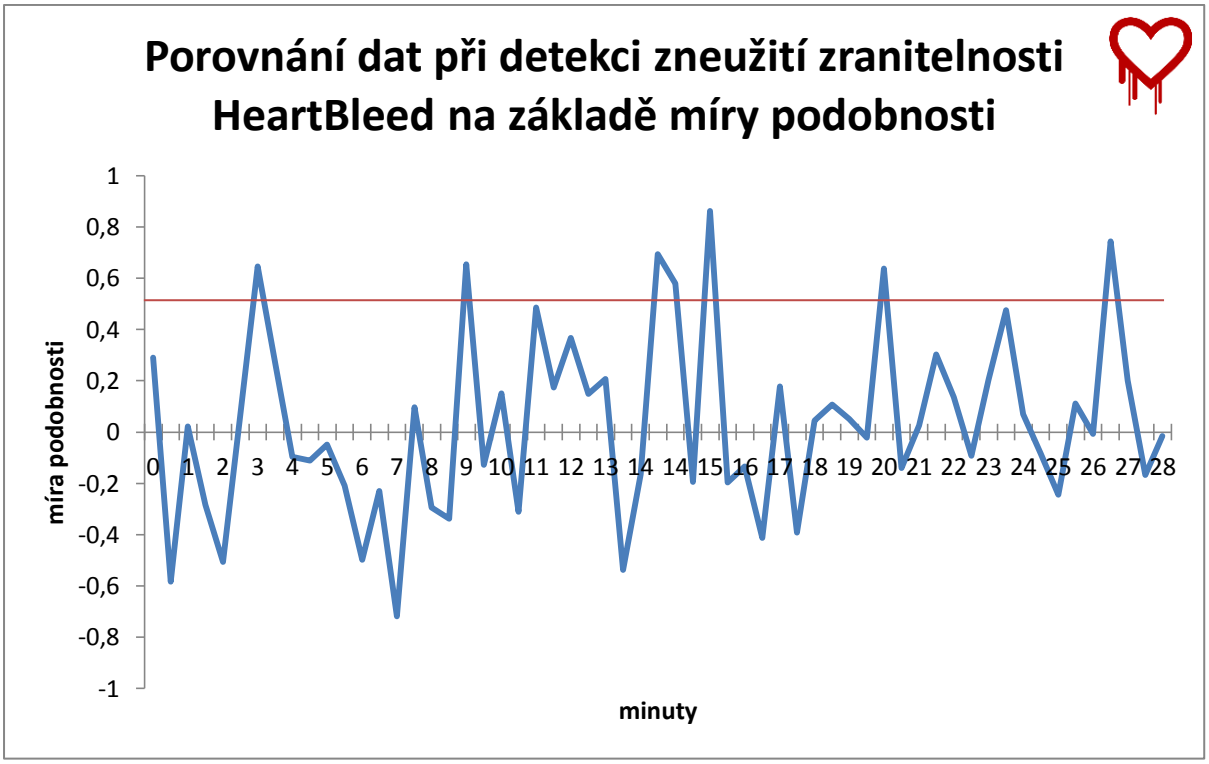
Po prozkoumání entropie v analyzovaném souboru (Obrázek 17) lze konstatovat, že od 3. do 25. minuty je entropie o 8 % nižší než od 25. minuty dále (průměrně 1,5 vs. 1,63). To by sice potvrdilo původní myšlenku, že v tomto období nastala anomálie, ale vzhledem k tomu, že nárůst entropie v tomto období je výrazně vyšší u referenčních dat než pokles entropie u analyzovaných dat, zastávám názor, že anomálie nastala u referenčních dat.

6.2 Zranitelnost HeartBleed

Chyba knihovny OpenSSL označovaná jako HeartBleed byla otestována také na dvou sadách dat. Obě sady sdílí stejná referenční data o délce 30 minut zachycená 4. 4. 2014 v dopoledních hodinách. Protože se nejedná o útok na systém DNS, před testy došlo k úpravě zdrojových kódů aplikace, aby pro výpočty zahrnula nejen DNS data, ale všechny IP toky.

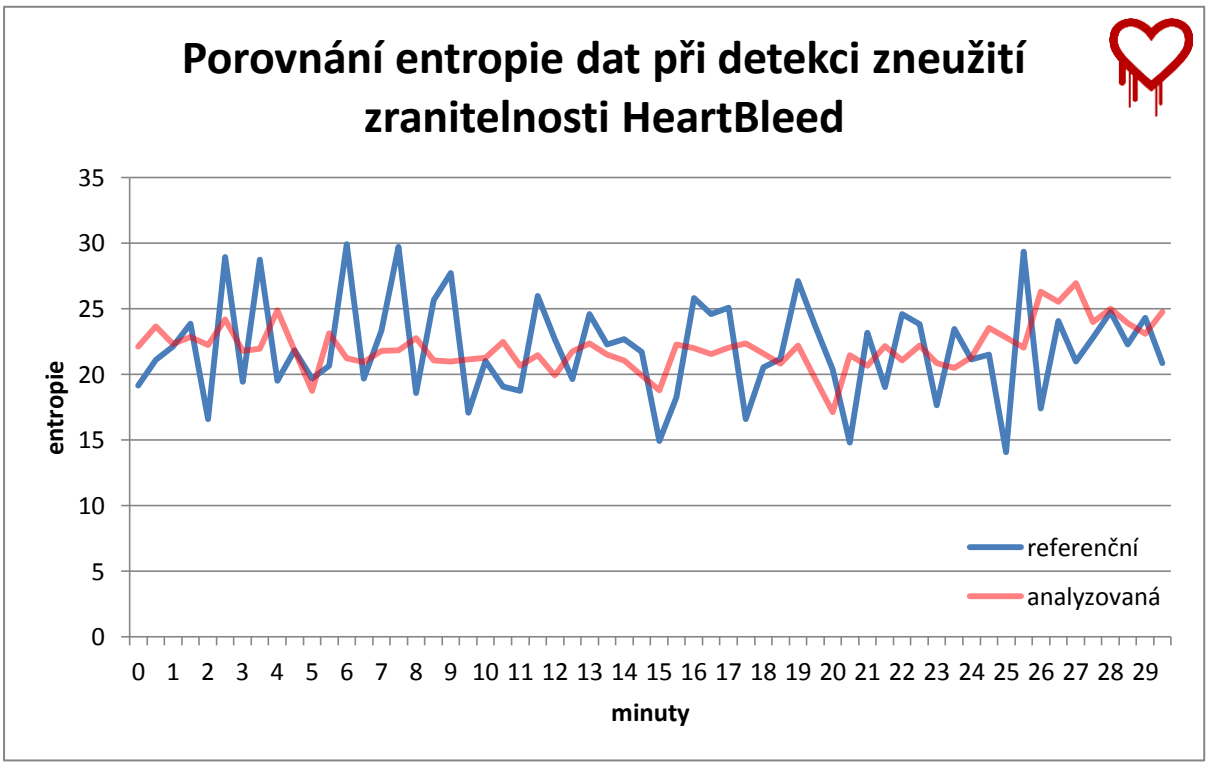
6.2.1 První sada dat

Analyzovaná data o délce 30 minut pochází z 25. 4. 2014. Metoda podobnosti při porovnání referenčních a analyzovaných dat vyhodnotí data jako nepodobná, viz Obrázek 18. Práh byl překročen pouze u 17 % vzorků, tedy 83 % půlminutových intervalů analyzovaných dat není podobných referenčním datům. Průměrná hodnota vychází na 0,0312.



Obrázek 18: Porovnání dat při detekci zneužití zranitelnosti HeartBleed na základě míry podobnosti (test 3)

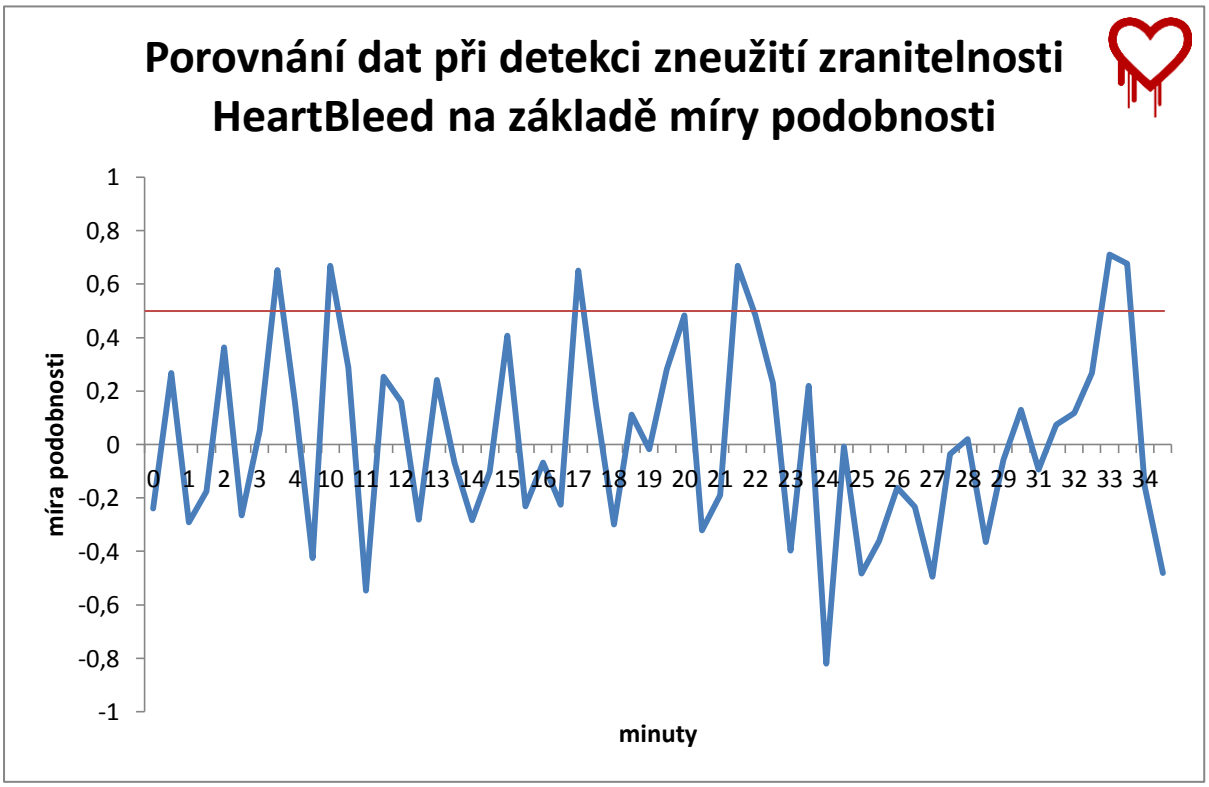
Obrázek 19 ukazuje graf entropie referenčních a analyzovaných dat. Z pohledu referenčních dat je entropie analyzovaných v rozpětí minimální a maximální entropie referenčních dat a žádná anomálie tak zřejmě nenastala. Budeme-li se ale dívat pouze na samotná analyzovaná data, jde si všimnout, že od 26. do 28. minuty entropie převyšuje do té doby nejvyšší amplitudu (o 8 %). Průměrná hodnota je v té době vyšší dokonce o 15 %. Pokud lze některý interval prohlásit za anomálii, tak to bude právě tento.



Obrázek 19: Porovnání entropie dat při detekci zneužití zranitelnosti HeartBleed (test 3)

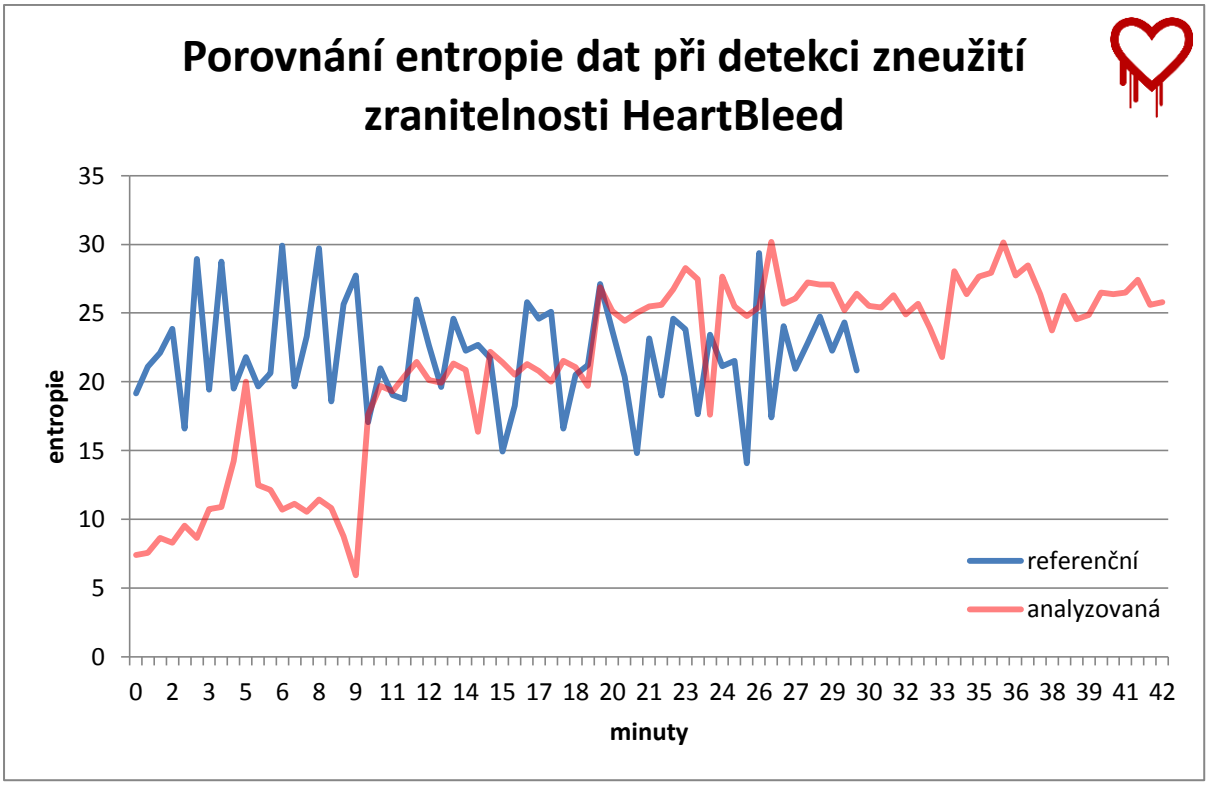
6.2.2 Druhá sada dat

Analyzovaná data o délce 40 minut byla zaznamenána o 75 minut později než analyzovaná data z první sady. Protože jsou o 10 minut delší než referenční, v metodě podobnosti bylo otestováno pouze prvních cca 30 minut analyzovaných dat.



Obrázek 20: Porovnání dat při detekci zneužití zranitelnosti HeartBleed na základě míry podobnosti (test 4)

Míra podobnosti (Obrázek 20) má podobný průběh jako v předchozím testu – většinu času je pod minimálním prahem, dokonce ve stejném počtu vzorků. Průměrná hodnota vyšla 0,0103. Podle této metody si tedy data podobná nejsou.



Obrázek 21: Porovnání entropie dat při detekci zneužití zranitelnosti HeartBleed (test 4)

V prvních deseti minutách analyzovaných dat je entropie výrazně pod minimální entropií referenčních dat. Takto znatelný rozdíl muselo způsobit hodně odlišné složení provozu na síti, takže toto období lze určitě považovat za anomálii. Ve zbývající půlhodině nelze říci s jistotou, nicméně 10. až 20. minuta vzhledem k posledním 20 minutám, má entropii také relativně nízkou, ale přesto se pohybuje v intervalu daným referenčními daty.

6.3 Zhodnocení výsledků

V této práci byly metody otestovány nejprve na detekci reflektivního DNS útoku. Metoda podobnosti měla nastaveny ideální podmínky pro výpočet korelačního koeficientu, referenční data v obou testech totiž pocházela ze stejné linky z doby před nebo po útoku. Princip metody byl také dodržen, protože NetFlow data ukládají počty paketů. Nicméně výsledky metody pro tento typ útoku přesvědčivé vůbec nejsou, protože podle nich se anomálie vyskytly po většinu času, což je nepravděpodobné.

Zato metoda entropie se osvědčila v obou testech. Očekával jsem sice výraznější nárůst entropie, její průběh ale závisí na síle útoku. Čím je útok silnější, tím více by měl ovlivnit entropii. Po vynesení dat do grafu metoda dokonce odhalila anomálie i v referenčních datech. To je ale problém pro současnou aplikaci této metody, tedy jako analýzu na základě porovnání s referenčními daty. Jak se ukázalo, oba referenční vzorky jsou totiž zatíženy anomáliemi. Je tedy velmi pravděpodobné, že zachytit správný referenční vzorek dat nebude vůbec jednoduché. Vzorek totiž nejen, že nesmí obsahovat anomálie, ale také nesmí být příliš starý, aby složení toků v něm zachycených přibližně odpovídalo analyzovaným datům. Naštěstí tato metoda nevyžaduje referenční data ve stejné délce jako analyzovaná, takže řešením by bylo analyzovat referenční data a extrahovat z nich tu část, kde se skutečně žádná anomálie nenachází a pouze tu použít jako referenční. Tato skutečnost ale jasně dokázala, jak je důležité mít prostředky pro detekci síťových anomálií.

Na datech nesouvisejících s DNS anomálií metoda podobnosti vykazuje podobné výsledky jako na reflektivní DNS útoku, takže to vypadá, že je tato metoda pro NetFlow data spíše

nepoužitelná. Ovšem metoda entropie má zajímavé výsledky. Na prvním vzorku dat sice samotný nástroj žádnou anomálii nedetekoval, protože se entropie analyzovaných dat pohybuje v intervalu daným referenčními daty, ale při bližším prozkoumání pouze na základě analyzovaných dat je patrno v poslední části viditelné zvýšení entropie poukazující na anomálii. V případě druhého testu jsou výsledky ještě průkaznější.

Práce se při testování zaměřila především na reflektivní DNS útok, to ale není jediná DNS anomálie. V případě útoku DNS cache poisoning by zřejmě opět záleželo na jeho síle, protože obě tyto metody pracují pouze se statistickými údaji o provozu. Pokud by byl útok příliš slabý, vůbec by se neprojevil u žádné z metod. V případě silnějšího útoku by se měl u metody entropie projevit nárůstem entropie zdrojových IP adres. Jak již bylo řečeno, tyto metody pracují pouze se statistickými údaji o provozu, jsou tedy schopné odhalit pouze ty útoky, při kterých dojde k výraznější změně charakteru provozu, např. tím, že někdo začne silně útočit z jedné IP adresy (nezávisle na typu útoku), dojde k výraznému nárůstu počtu paketů na síti, což by měla odhalit metoda podobnosti, a zároveň zdrojová IP adresa, popř. port, bude tvořit významnější část provozu, což se projeví na změně entropie zdrojové IP adresy, popř. portu. Situace bude obdobná, pokud bude útok distribuovaný, ale cílit bude na úzkou skupinu IP adres či portů. Například škodlivé domény tímto způsobem detekovat nelze vůbec, protože metody nepočítají s daty, ze kterých by to zjistit šlo. V případě NetFlow dat pak metody ta data ani k dispozici nemají.

Dalším výzkumem by bylo možné otestovat metodu entropie pro detekci jiných typů útoků nejen u DNS provozu. Například u DoS či DDoS útoků by metoda mohla být úspěšná, protože z principu metody je jedno, zda testuje DNS či jakákoliv jiná data.

7 Závěr

Cílem této bakalářské práce bylo implementovat aplikaci, která bude v zachyceném síťovém provozu detekovat DNS anomálie na základě referenčních dat pomocí metody podobnosti a entropie.

V úvodu byla rozebrána hlavní funkce a cíl služby DNS z pohledu běžného uživatele internetu, důležitost udržení této služby v provozu, důsledky její nefunkčnosti a zmíněna nejčastější metoda zabezpečení integrity dat v tomto systému.

Další kapitola se plně věnuje systému DNS, principu jeho hierarchickému návrhu doménových jmen, způsobu rezoluce dotazu, typům záznamů a jejich uložení na DNS serverech. Dále také popisuje nejčastější metodu získávání síťových dat pomocí NetFlow exportérů a celý princip protokolu NetFlow včetně popisu NetFlow záznamů.

Následuje důležitá část práce, která vysvětluje pojem DNS anomálie, její jednotlivé typy, projevy, principy zneužití a důsledky výskytu. Také se podrobně zaměřuje na vysvětlení principu metod podobnosti a entropie a způsobů jejich použití pro detekci anomálií.

V dalších částech je nejprve navržen princip činnosti nástroje, a pak jsou rozebrány detaily implementace čtení zdrojových dat včetně základního popisu použitých knihoven. Několik následujících odstavců komentuje implementaci samotných metod a možnosti, jak interpretovat výstup z aplikace.

Poslední kapitola se věnuje testování aplikace na několika vzorcích anonymizovaných dat pocházejících ze sítě FIT VUT v Brně. Nejprve byly provedeny testy na detekci reflektivního DNS útoku a dále byly obě metody ověřeny, zda je lze aplikovat i na jiný typ dat, než jen DNS, konkrétně na zranitelnost HeartBleed. Výsledky testování byly detailně analyzovány a důležitým zjištěním bylo, že anomálie se v síťovém provozu vyskytují často, dokonce se mohou vyskytnout i u referenčních dat. Naštěstí díky principu metody podobnosti je možné je detekovat i v referenčním provozu. Výsledky také ukazují, jak je důležité být informovaný o těchto anomáliích. Velmi zajímavým rozšířením by tak mohla být úprava nástroje pro detekci anomálií v reálném čase.

Literatura

[1] „Domény podle DNSSEC,“ CZ.NIC, z.s.p.o., 21 únor 2014. [Online]. Dostupné z: http://stats.nic.cz/stats/domains_by_dnssec/. [Přístup získán 22 únor 2014].

[2] P. Mutton, „Half a million widely trusted websites vulnerable to Heartbleed bug | Netcraft,“ 8 duben 2014. [Online]. Dostupné z: http://news.netcraft.com/archives/2014/04/08/half-a-million-widely-trusted-websites-vulnerable-to-heartbleed-bug.html. [Přístup získán 5 květen 2014].

[3] J. Kirk, „Critical OpenSSL 'Heartbleed' bug puts encrypted communications at risk,“ PCWorld, 8 duben 2014. [Online]. Dostupné z: http://www.pcworld.com/article/2140920/heartbleed-bug-in-openssl-puts-encrypted-communications-at-risk.html. [Přístup získán 2 květen 2014].

[4] R. Graham, „Errata Security: 300k servers vulnerable to Heartbleed one month later,“ 8 květen 2014. [Online]. Dostupné z: http://blog.erratasec.com/2014/05/300k-servers-vulnerable-to-heartbleed.html. [Přístup získán 9 květen 2014].

[5] Wang, Z.; Zhang, M.: The Research of DNS Anomaly Detection Based On The Method of Similarity And Entropy. V *International Conference on Intelligent Computation Technology and Automation*, IEEE, 2010, ISBN 978-0-7695-4077-1/10, s. 905-909.

[6] Wang, Z.; Zhang, M.: The Realization of DNS Anomaly Detection Based On Combination of Two Methods of Similarity And Entropy. V *Second International Conference on Computational Intelligence and Natural Computing (CINC)*, IEEE, 2010, ISBN 978-1-4244-7703-6/10, s. 113-116.

[7] B. Fajmon a I. Růžičková, *Matematika 3*, Brno, 2005.

[8] C. E. Shannon, „A Mathematical Theory of Communication,“ *The Bell System Technical Journal*, sv. 27, s. 379-423,623-656, červenec, říjen 1948.

[9] V. Jacobson, C. Leres a S. McCanne, „Manual page of PCAP,“ 1 červenec 2013. [Online]. Dostupné z: http://www.tcpdump.org/pcap3_man.html. [Přístup získán 15 prosinec 2013].

[10] B. Claise, *Cisco Systems NetFlow Services Export Version 9*, RFC 3954, 2004.

[11] P. Mockapetries, *Domain Names — Concepts and Facilities*, RFC 1034, 1987.

[12] P. Mockapetris, *Domain Names — Implementation and Specification*, RFC 1035, 1987.

[13] B. Claise, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*, RFC 7011, 2013.

[14] P. Matoušek, Síťové služby a jejich architektura, Brno: Nakladatelství Vysokého učení technického v Brně VUTIUM, 2014.

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

[16] L. M. Garcia, „Programming with Libpcap - Sniffing the Network From Our Own Application,“ *Hakin9 Magazine*, sv. 3, č. 2, s. 38-46, 2/2008.

Seznam obrázků

Seznam příloh

Příloha A. Obsah CD

Přiložený optický disk obsahuje zdrojové kódy a knihovny vytvořeného nástroje včetně souboru Makefile pro překlad na linuxových systémech a solution pro Microsoft Visual Studio 2010. Dále na něm naleznete soubor README s detailnějším popisem adresářové struktury, binární verze nástroje a také tuto technickou zprávu ve formátu PDF a zdrojového souboru.

Příloha B. Požadavky na OS

Implementovaný nástroj byl testován v prostředích linuxové distribuce Debian ve verzi 7.1 (32-bit) a 7.4 (64-bit). Všechny zdrojové kódy jsou umístěny na přiloženém optickém disku. Aplikace vyžaduje knihovny nfreader a libpcap (pokud překládáte na linuxu) nebo WinPcap (pokud překládáte na Windows). Knihovna nfreader je přiložena (ve verzi pro 32- i 64-bitovou architekturu OS GNU/Linux), libpcap / WinPcap je potřeba nainstalovat před samotným překladem aplikace. Překlad lze provést na linuxu pomocí aplikace g++, na Windows nejlépe pomocí aplikace Microsoft Visual Studio pomocí přiloženého solution.

Příloha C. Uživatelská příručka

Implementovaný nástroj neobsahuje grafické rozhraní, jedná se o tzv. konzolovou aplikaci, tzn., že se ovládá z příkazového řádku.

Aplikace se ovládá pomocí následujících parametrů:

`-r <soubor>` relativní nebo absolutní cesta k souboru s referenčními daty (povinný)

`-f <soubor>` relativní nebo absolutní cesta k souboru s analyzovanými daty (povinný)

`-d <složka>` pokud je uveden, pak aplikace do zadané složky ukládá pomocná data, ze kterých lze vytvořit grafy podobné těm v kapitole Testování.

Vstupní data aplikace mohou být pro platformu linux ve formátech PCAP a NetFlow, pro Windows ve formátu PCAP.

Pomocná data generovaná aplikací při zadání parametru `-d` jsou ve formátu CSV (Comma-separated values, hodnoty oddělené čárkami; v tomto případě středníky, protože čárka v českém národním prostředí odděluje desetinná místa). V podstatě se jedná o sloupce sešitu např. pro aplikace Microsoft Excel či OpenOffice. Celkem jsou vygenerovány 3 soubory.

Soubory *entropy_reference.csv* a *entropy_analyzing.csv* obsahují vypočítanou entropii pro referenční, resp. analyzovaný soubor. Jednotlivé sloupce obsahují časovou značku (timestamp), celkovou entropii, entropii ze zdrojových IP adres, cílových IP adres, zdrojových portů, cílových portů a délek paketů. Soubor *similarity.csv* obsahuje dva sloupce, časovou značku a míru podobnosti.

Protože aplikace přijímá pouze jeden analyzovaný a jeden referenční soubor a v praxi tyto soubory obsahují třeba jen 5 minut dat, je vhodné automaticky analyzovat celou složku obsahující tyto soubory. Součástí aplikace je tedy také skript *test_folder.sh*, který se o to postará. Skript přijímá jako první parametr cestu ke složce s referenčními soubory a jako druhý cestu ke složce s analyzovanými soubory. Skript automaticky vytváří soubory jako aplikace při zadání parametru `-d`. Tyto soubory agreguje ze všech vstupních souborů daných složek, ve výsledku tedy vytvoří opět pouze 3, resp. 4 soubory. Všechny soubory se vytvoří v nadřazených složkách zadaných složek. V nadřazené složce složky s referenčními soubory vzniknou dva soubory, *entropy_{název-referenční-složky}.csv* a *similarity_{název-referenční-složky}_{název-analyzované-složky}.csv*. Stejně tak vzniknou soubory v nadřazené složce složky s analyzovanými soubory. Skript při svém běhu vytváří dočasné soubory ve složce /tmp (přesněji řečeno se tam ukládají pomocné soubory generované samotnou aplikací).