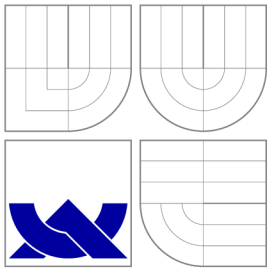
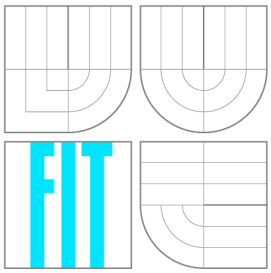


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ANALÝZA SÍŤOVÉHO PROVOZU POMOCÍ SHLUKOVÉ ANALÝZY

NETWORK TRAFFIC ANALYSIS BASED ON CLUSTERING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ ČERNÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VÁCLAV BARTOŠ

BRNO 2013

Abstrakt

Tato práce se zabývá detekcí anomálií v síťovém provozu pomocí shlukové analýzy. V úvodu je popsáno základní rozdělení metod detekce anomálií s jejich krátkým popisem. Následně jsou detailněji popsány metody hierarchického a k-means shlukování a vybrané techniky normalizace. Část je také věnována postupu při detekci anomálií v kontextu dolování dat. Dále je popsána implementace jednotlivých metod. Další část tvoří vyhodnocení metod a jejich vzájemné porovnání a vyvození závěrů.

Abstract

This thesis focuses on anomaly detection in network traffic using clustering methods. First, basic anomaly detection methods are introduced. The next part discribes hierarchical and k-means clustering in detail. Also there are described selected normalization techniques. Part is given to the procedure for detecting anomalies in the context of data mining. Furthermore a few words about implementation of single methods. Finally, clustering methods and normalization techniques are tested and compared.

Klíčová slova

Bezpečnost sítí, Detekce anomálií, Shluková analýza, Hierarchické shlukování, K-means

Keywords

Network security, Anomaly detection, Clustering, Hierarchical clustering, K-means

Citace

Tomáš Černý: Analýza síťového provozu pomocí shlukové analýzy, bakalářská práce, Brno, FIT VUT v Brně, 2013

Analýza síťového provozu pomocí shlukové analýzy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Bartoše.

.....
Tomáš Černý
13. května 2013

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Václavu Bartošovi za jeho cenné rady a připomínky. Také za čas, který mi věnoval při konzultacích k této práci, a přístup k testovacím datům.

Obsah

Kapitola 1

Úvod

Internet a počítačové sítě zažívají v současné době velký růst. Roste počet organizací využívající sílu počítačových sítí, i běžných uživatelů. S neustálým vývojem se však zvyšuje i počet útoků a obecně bezpečnostních hrozeb, před kterými se je nutné chránit.

Firewall, který je považován za typický ochranný prvek, poskytuje pouze základní zabezpečení. Pro zajištění kvalitního zabezpečení je nutné použít další technologie, které jsou schopné detekovat sofistikovanější útoky (například DoS útoky na webové služby). Jednou z technologií jsou systémy pro odhalení průniků (*Intrusion Detection Systems*, IDS).

IDS systémy monitorují síťový provoz a detekují podezřelé chování. Tyto systémy obecně nezasahují a nesnaží se útok přerušit (jsou pasivní), ale mají za úkol varovat a zaznamenávat. Naopak aktivní systémy, které proti detekované události zasahují, jsou známé jako systémy prevence proniknutí (*Intrusion Prevention Systems*, IPS). Detekci lze rozdělit do dvou hlavních kategorií: detekce signatur a detekce anomálií.

U systémů založených na detekci signatur je nutná databáze vzorů, vůči které je porovnáván aktuální provoz na síti. V případě nalezení shody je hlášen útok (je k dispozici i informace o jaký útok se jedná). Pro správný chod je nutné udržovat databázi vzorů aktuální (obsahuje signatury všech známých útoků). Tyto systémy nejsou schopné detekovat nové typy útoků, ale generují nízký počet falešných poplachů. Lze tedy tyto systémy spojit přímo s IPS systémy, aniž by se musela provádět podrobnější analýza útoku.

Systémy detekující anomálie pracují s normálním chováním sítě, na jehož základě jsou pak schopné detekovat anomálie. Pro klasifikaci, zda se jedná o normální chování nebo anomálii, se využívá různých heuristických a matematických funkcí. Mimo jiné se využívá například principů strojového učení a dolování dat. Tyto systémy jsou schopné detekovat nové, neznámé útoky (tzv. *zero days* útoky), ale generují vyšší počet falešných poplachů. Proto je tedy obecně nutné takto detekované události blíže prozkoumat a potvrdit nebo vyvrátit hrozbu (například zásah správce sítě).

Oba dva systémy lze zkombinovat a vytvořit tak systémy hybridní, které kombinují jejich vlastnosti tak, aby se dosáhlo co nejlepšího výsledku pro danou síť.

Cílem práce je srovnání dvou metod shlukové analýzy a několika typů normalizací při detekci anomálií nad daty ze síťového provozu. V následující kapitole 2 je popsán jeden ze způsobů získávání informací o síťovém provozu. Kapitola 3 obsahuje základní informace o detekci anomálií a jejich dělení. Další kapitola 4 je zaměřená na detekci anomálií pomocí shlukové analýzy. Jsou zde popsány metody hierarchického a k-means shlukování, část je věnována postupu při detekci anomálií v kontextu dolování dat. Kapitola 5 popisuje, jakým způsobem jsou metody implementovány a jakým způsobem je možné detekci otestovat. V kapitole 6 jsou jednotlivé metody a různé techniky normalizace otestovány a jsou prezentovány výsledky a závěry. Na závěr 7 jsou shrnuty získané poznatky a je zrekapitulována celá práce.

Kapitola 2

NetFlow

Zachytávání síťových toků patří dnes k nejběžnějšímu způsobu monitorování sítí, kromě NetFlow protokolu, který bude v této kapitole blíže popsán, existují i jiné variatny pro sběr informací o síťovém provozu. Data používaná v této práci nejsou přímo NetFlow záznamy, ale jsou z NetFlow záznamů vytvořena, a proto je vhodné jednu kapitolu věnovat právě NetFlow protokolu.

NetFlow je protokol vyvinutý firmou Cisco a je určen pro přenos záznamů, obsahující informace o síťových tocích. Protokol byl původně určen pro Cisco směrovače, kde sloužil jako doplňková služba pro monitorování sítě. Protokol je uzavřený, ale je k dispozici jeho specifikace v RFC 3954 [1], díky níž se stal NetFlow protokol velmi rozšířený a hojně používaný.

Síťový tok, který je základem NetFlow záznamu, je definován jako sekvence paketů s několika stejnými vlastnostmi. Stejnými vlastnostmi se v kontextu NetFlow rozumí shodná zdrojová a cílová IP adresa, zdrojový a cílový port, IP protokol, typ služby (anglicky Type of Service, ToS) a číslo rozhraní. Společně s dalšími užitečnými informacemi o toku (viz. sekce 2.2.1) jsou NetFlow záznamy mocným prostředkem pro monitorování sítí a jsou například využívány systémy pro detekci anomálií pro zajištění vyššího zabezpečení.

2.1 Architektura NetFlow

Architektura NetFlow se skládá ze dvou základních prvků:

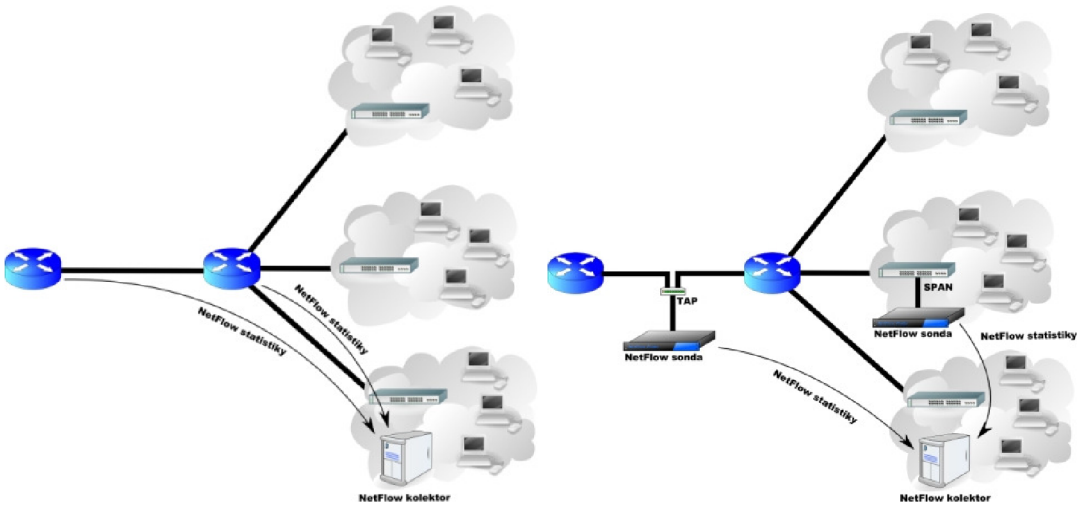
- **Exportér** – Exportér monitoruje síť a vytváří záznamy o tocích z paketů, které jím procházejí. Záznam o toku je v exportéru vytvořen při zachycení paketu, který nepatří do žádného existujícího toku. Tok je považován za ukončený, pokud bylo přerušeno TCP spojení (příznak FIN nebo RST) nebo došlo k vypršení časovačů (neaktivní tok, příliš dlouhý tok), případně pokud hrozí zaplnění paměti nebo přetečení čítačů. Tyto záznamy o ukončených tocích jsou následně odesílány v podobě NetFlow paketů kolektorům. Každý NetFlow paket může obsahovat jeden a více záznamů.

- **Kolektor** – Kolektor přijímá a zpracovává pakety odeslané exportéry a ukládá získaná data na disk. Nad těmito daty je možné vytvářet ruzné statistiky (nástroj `nfdump`), přehledné grafy či tabulky (nástroj `nfsen`) a další.

Typicky se v monitorované oblasti nachází jeden a více exportérů, které odesílají pakety do jednoho centrálního kolektoru.

Při vzniku NetFlow protokolu byly exportéry součástí Cisco směrovačů, kterým tak kromě vlastních povinností přibyla i nutnost zpracovávat jednotlivé toky. To znamenalo zvýšení nároků na výkon a tedy i zvýšení ceny směrovačů. Jako částečné řešení problému se využívá vzorkování, kdy se pro vytváření záznamů o tocích zpracovává každý n-tý paket. To má za následek snížení přesnosti statistik a také snížení pravděpodobnosti odhalení bezpečnostního rizika.

Proto se v dnešní době stále častěji využívá speciálních hardwarových zařízení, takzvaných NetFlow sond. Jediným úkolem těchto sond je monitorování a export záznamů o NetFlow tocích, což je činí levnější a výrazně výkonější v porovnání se směrovači podporujícími NetFlow protokol. Sondy NetFlow jsou pasivní zařízení, která žádným způsobem nezasahují do procházejících paketů. V případě použití dedikovaných linek pro export záznamů jsou sondy zcela transparentní, neovlivňují provoz na sítí a je velmi obtížné proti nim podniknout útok.



Obrázek 2.1: Vlevo tradiční architektura se směrovači, vpravo moderní architektura s HW sondami [12].

2.2 NetFlow verze

Nejrozšířenější verze protokolu NetFlow jsou verze 5 a 9. Verze 5 má pevně danou strukturu paketu a podporuje pouze IPv4. Verze 9 se liší tím, že podporuje šablony, které umožňují měnit strukturu paketu a tedy i informace, které budou zaznamenávány a přenášeny. Verze 9 navíc podporuje i IPv6.

Z nejnovější verze vychází protokol IPFIX (*IP Flow Information Export*), který se stal protokolem IETF. Byl vytvořen za účelem získání veřejného a univerzálního standardu pro export informací o síťových tocích. Základní specifikace je k dispozici v RFC 5101 [2].

2.2.1 Informace obsažené v NetFlow záznamech

Protokol NetFlow ukládá nejrůznější informace o tocích, které lze získat a zobrazit například pomocí nástroje `nfdump`. Příklad informací, které lze získat a s jejichž podmnožinou se bude dále v této práci pracovat [4]:

Start Time	čas, kdy byl tok porvé zaznamenán
End Time	čas, kdy byl tok ukončen
Duration	délka toku
Protocol	použitý protokol
Source/Destination Address	zdrojová a cílová adresa
Source/Destination Port	zdrojový a cílový port
Source/Destination AS	zdrojový a cílový autonomní systém
Input/Output Interface num	číslo vstupního a výstupního rozhraní
Packets	počet paketů v toku
Bytes	počet bytů v toku
Flows	počet toků (při agregaci více toků)
TCP Flags	příznaky u TCP paketů (sjednocení všech příznaků, které se během toku vyskytly)
ToS	typ služby
bps	počet bitů za sekundu
pps	počet paketů za sekundu
bpp	počet bytů za paket

Kapitola 3

Detekce anomálií

Systém pro detekci anomálií funguje na principu nacházení činností (aktivit), které se nějakým způsobem odlišují od normálního chování v systému. Takto detekované aktivity jsou označené jako potencionální hrozba. Nelze totiž s jistotou tvrdit, že jde o škodlivou činnost.

Použití systémů pro detekci anomálií jako zabezpečení sítě má jisté výhody. Oproti systémům založených na detekci signatur umožňují tyto systémy odhalit doposud neznámé útoky. Další výhodou je určitý stupeň flexibility, kdy může systém pracovat v různých režimech s různým nastavením parametrů (denní/noční režim, různé nastavení prahů). Pro útočníka je pak těžší odhadnout, zda jeho pokus bude interpretován jako anomálie či nikoliv. Flexibility je dosaženo i tím, že systém pracuje s aktuálním děním na síti, avšak toto lze chápat i jako nevýhodu. Například pokud bude v danou dobu na síti převládat škodlivá činnost, může to být systémem paradoxně chápáno jako normální chování a naopak minoritní běžný provoz může být nahlášen jako útok.

3.1 Předpokládané výsledky detekce anomálií

Při zpracovávání výsledků detekce anomálií je typické, že množina škodlivých aktivit je podmnožinou podezřelé (anomální) aktivity. V ideálním případě je veškerá aktivita označená jako anomální zároveň škodlivá – pak jde o výsledek s nulovými falešně pozitivními a falešně negativními detekcemi. Škodlivá aktivita ovšem nemusí být vždy součástí množiny anomálií. Běžně se rozlišují čtyři typy výsledků, které mohou při detekci nastat [9]:

- **Falešně negativní (*false negatives*)**: případ, kdy systém selže v detekování škodlivé činnosti a neoznačí útok jako anomální činnost. Počet falešně negativních výskytů lze minimalizovat snížením hodnot použitých prahů u detekčních algoritmů, ovšem za cenu zvýšení počtu falešně pozitivních výskytů.
- **Falešně pozitivní (*false positives*)**: aktivita označená jako anomální, která ale není žádným způsobem škodlivá.
- **Pravé negativní (*true negatives*)**: aktivita není označená jako anomální a zároveň se nejedná o škodlivou činnost.
- **Pravé pozitivní (*true positives*)**: správné označení škodlivé činnosti jako anomální.

3.2 Přehled metod detekce anomálií

Existuje mnoho metod a technik zabývajících se detekcí anomálií, kde každá má své výhody a nevýhody. Většina metod musí nejdříve projít tzv. trénovací fází, od které se následně odvíjí samotná detekce anomálií. V trénovací fázi je vytvořen model normálního chování sítě, nejlépe na základě dat, která odráží aktuální dění na síti, bez anomálního (škodlivého) provozu. Ná základě kvality normálního modelu sítě je pak závislá efektivita detekce anomálií v běžném provozu. Fáze, kdy probíhá odhalování potencionálních útočníků, se nazývá testovácí. Všechny metody lze zařadit podle principů, ze kterých vycházejí, do několika kategorií: statistické metody, strojové učení a dolování dat [9] [11].

Statistické metody typicky udržují dva modely chování. První model lze považovat jako referenční, který byl vytvořen na základě dat dříve získaných. Tento je porovnáván s modelem aktuálním a při výrazných odchylkách aktuálního modelu od referenčního je detekována anomálie. Pro detekci se využívá funkcí abnormality, které porovnáním obou modelů určí hodnotu skóre anomality a porovná je s nastavenými prahy. Referenční model se s postupem času aktualizuje modelem aktuálním.

Strojové učení, odvětví umělé inteligence, lze definovat jako schopnost programu učit se a zlepšovat se v daném úkolu postupem času. Systémy založené na strojovém učení jsou tak schopné na základě nově získaných dat měnit svoje chování a detekční schopnosti.

Oblast dolování dat (*data mining*) má velmi blízko ke strojovému učení a v některých zdrojích jsou tyto dvě oblasti spojovány do jedné. Dolování dat pracuje s velkým množstvím dat, ze kterých je schopno získat skryté a potencionálně užitečné informace. Někdy je dolování dat označováno termínem *knowledge discovery*.

Shluková analýza (*clustering*) zasahuje z části jak do strojového učení, tak do oblasti dolování dat. Shluková analýza vytváří shluky sobě navzájem podobných dat, které jsou později v kontextu detekce anomálií v počítačových sítích označené jako normální nebo anolmální. Anomálie mohou vytvořit buď menší anomální shluky, nebo jsou natolik odlišné od ostatních, že nejsou klasifikovány do žádného shluku a jde o takzvané *outliers* (viz. další kapitola 4).

Kapitola 4

Detekce anomálií pomocí shlukové analýzy

Cílem shlukové analýzy je nalezení (vytvoření) několika skupin dat (shluků), kdy vlastnosti objektů, které tyto shluky tvoří, jsou stejné nebo do určité míry podobné a zároveň vlastnosti objektů z různých shluků jsou znatelně odlišné. Shluková analýza se hojně využívá pro analýzu vícedimenzionálních dat v nejrůznějších oborech. Velkou výhodou shlukové analýzy je, že se může učit přímo z pozorovaných dat – nepotřebuje tedy projít počáteční trénovací fází.

Vzájemná podobnost (similarita) objektů je klíčovou vlastností pro shlukovou analýzu. Na základě toho jak jsou si objekty podobné, jsou klasifikovány do příslušného shluku. Hodnota podobnosti je často založená na vzájemné vzdálenosti, konkrétně na vzdálenosti Euklidovské, ale existuje mnoho jiných možností a algoritmů k výpočtu podobnosti. Euklidovskou vzdálenost `d` mezi objekty `x` a `y` v `n`-rozměrném prostoru lze vypočítat jako:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

V rámci shlukové analýzy lze narazit i na termín *outliers*. *Outliers* jsou objekty, které nenáleží do žádného shluku a mohou v kontextu detekce anomálií představovat útoky.

Důvodem proč existuje velké množství algoritmů pro shlukovou analýzu je také ten, že se je třeba přizpůsobit nejrůznějším typům dat. V této práci shlukovaná data (jednotlivé objekty) vychází ze statistik síťových hostů získaných ze záznamů NetFlow. Pro každého hosta je vytvořen vlastní vektor vlastností, se kterým následně shlukovací algoritmy pracují. Více informací o vektorech vlastností použitých v této práci lze nalézt v kapitole 4.3.1.

4.1 Metody shlukové analýzy

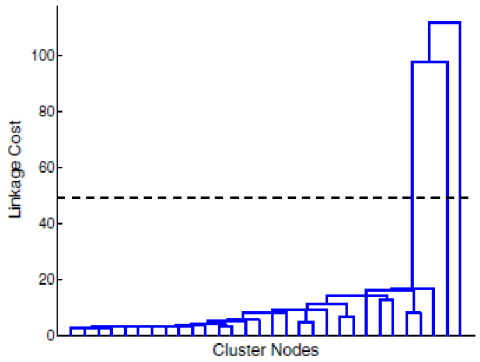
Algoritmů shlukové analýzy a jejich variant existuje velké množství. Je to dáno také tím, že se tyto metody využívají v širokém spektru oborů. Na základě toho, jak dané metody pracují a z jakých principů vycházejí, je lze rozdělit do několika hlavních skupin: hierarchické metody (*hierarchical*), metody založené na dělení (*partitioning*), *density-based*, *model-based* a *grid-based* metody. Vzhledem k rozsáhlosti této problematiky zde budou popsány pouze první dvě kategorie, jejichž principy jsou v této práci použity pro detekci anomálií. Informace v této sekci jsou čerpány ze zdrojů [7] a [10].

4.1.1 Hierarchické metody

Tyto metody vytváří shluky postupným spojováním nebo rozpojováním objektů na základě podobnosti podle toho, zda jde o přístup shora-dolů nebo zdola-nahoru. Dělí se tedy na dvě podskupiny: aglomerativní (*aglomerative*) a dělící (*divisive*) hierarchické shlukování.

- **Aglomerativní hierachické shlukování (přístup zdola-nahoru)** – Na počátku je každý objekt umístěn do vlastního shluku. Tyto jsou následně spojovány do větších shluků, dokud nejsou nakonec součástí jediného hlavního shluku nebo dokud není splněna jiná podmínka, která spojování v určité fázi ukončí.
- **Dělící hierachické shlukování (přístup shora-dolů)** – Nejprve jsou všechny objekty součástí jednoho shluku, který je poté dělen do shluků menších, až nakonec každý objekt reprezentuje svůj vlastní shluk.

Výsledkem hierarchického shlukování je dendrogram, reprezentující vzniklou stromovou hierarchii (viz. obrázek 4.1). Uzly stromu jsou jednotlivé shluky, přičemž kořenem je shluk obsahující všechny objekty, listy stromu pak reprezentují přímo jednotlivé objekty. Kromě stromové hierarchie dendrogram také znázorňuje, s jakými hodnotami podobnosti (rozdílnosti) dané shluky vznikly.



Obrázek 4.1: Dendrogram zobrazující výsledný strom hierarchického shlukování [3].

Kromě volby funkce pro výpočet podobnosti se navíc u hierarchického shlukování musí specifikovat i způsob jak se bude podobnost mezi jednotlivými shluky počítat – jde o takzvaná linkovací kritéria (*linkage criterion*). Mezi základní způsoby patří:

- **Single-linkage** – Také známé pod názvy *connectedness*, minimální metoda nebo metoda nejbližšího souseda. Vzdálenost (podobnost) mezi dvěma shluky je rovna minimální vzdálenosti (maximální podobnosti) mezi libovolným objektem z prvního shluku a libovolným objektem z druhého shluku. Použití této metody má efekt řetězení objektů – i velmi málo objektů může způsobit vytvoření "mostu", který spojí dva shluky do jednoho.
- **Complete-linkage** – Také známé pod názvy *diameter*, maximální metoda nebo metoda nejvzdálenějšího souseda. Vzdálenost (podobnost) mezi dvěma shluky je rovna maximální vzdálenosti (minimální podobnosti) mezi libovolným objektem z prvního shluku a libovolným objektem z druhého shluku. *Complete-linkage* metody obecně

produkují kompaktnější shluky a užitečnější hierarchie, ale *single-linkage* metody jsou považovány za více univerzální.

- **Average-linkage** – Metoda minimálního rozptylu. Vzdálenost mezi dvěma shluky je rovna průměrné vzdálenosti mezi objekty z jednoho shluku a objekty z druhého shluku.
- **Centroid-linkage** – Vzdálenost shluků je dána vzdáleností center těchto shluků.

Výše zmíněné linkovací metody mohou být ve variantě neváženě (*unweighted*) nebo váženě (*weighted*). Vážené varianty se liší od nevážených tím, že se bere při výpočtu vzdálenosti ohled na počet objektů, které náleží do daného shluku.

Aglomerativní hierarchické shlukování

Jak již bylo zmíněno, tento algoritmus postupně sdružuje jednotlivé objekty do shluků až do fáze, kdy je každý objekt součástí jediného shluku. Algoritmus aglomerativního shlukování by mohl být popsán následovně:

1. Každý objekt umísti do svého vlastního shluku.
2. Vypočítej nové hodnoty v matici podobností.
3. Najdi takový pár, který si je nevíce podobný a spoj ho do nového shluku.
4. Uprav matici podobností smazáním páru objektů (shluků), který byl sloučen do nového shluku, a vytvoř nový záznam pro nově vzniklý shluk.
5. Pokud jsou všechny objekty součástí jednoho shluku skonči, jinak běž na krok 2.

Časová složitost aglomerativního shlukování je $O(M^2 * \log M)$ a paměťová náročnost je $O(M^2)$ z důvodu potřeby uchovat matici podobnosti, která má velikost právě M^2 (M vyjadřuje počet objektů).

Pro potřeby detekce anomálií je třeba rozhodnout, zda jsou objekty (hosté) v daném shluku potencionální útočníci nebo jsou součástí normálního síťového provozu. Je tedy třeba zajistit zastavení algoritmu dříve než budou všichni hosté součástí jediného shluku a tedy nebude žádná možnost, jak rozhodnout zda jde o anomálii, či ne. Jedním způsobem může být určení cílového počtu shluků, kdy se algoritmus zastaví, a následně na základě počtu hostů v daném shluku rozhodnout, zda jde o anomálii (počet hostů ve shluku je menší než hodnota zadaného prahu). Tento přístup ale potlačuje výhodu aglomerativního shlukování tím, že je potřeba předem zadat počet cílových shluků. Druhá možnost, která byla použita v [3], využívá hodnot similarit objektů určených ke sdružení do nového shluku. Pokud budeme uvažovat cenu za sjednocení dvou objektů během i-té iterace $L(i)$, která je během iterací neklesající, lze shlukování ukončit za splnění podmínky:

$$L(i) > \alpha * L(i-1), \quad \alpha > 1.$$

Při vhodně zvoleném parametru α (dále v práci označován i jako `alfa`) se shlukování ukončí při výraznější změně v podobnostech objektů. Poté už jen zbývá opět na základě počtu hostů ve shluku a zadaného prahu určit, zda jde o anomálie nebo ne.

4.1.2 Metody založené na rozdělování (*partitioning methods*)

Partitioning metody jsou pravděpodobně nejpoužívanější skupina algoritmů shlukové analýzy. Principem těchto metod je iterativní přerozdělování objektů mezi jednotlivými shluky. Objekty se během každé iterace přesouvají mezi shluky s cílem minimalizovat zadaná kritéria (například vzdálenosti mezi objekty v daném shluku). Iterace probíhají dokud se nedosáhne optimálního rozložení objektů v rámci shluků. Nejjednodušším a běžně používaným algoritmem je *k-means*, který bude popsán v následující části této kapitoly.

Nevýhodou těchto metod je nutnost určit počet cílových shluků před zahájením výpočtu. Nevhodně zvolený počet může způsobit zcela chybnou interpretaci vstupních dat. Další nevýhodou je, že nalezené optimální řešení je často optimální pouze lokálně, nikoli globálně.

K-means

Princip, na kterém je algoritmus k-means založen, je velmi jednoduchý a intuitivní. Je ale nutné zadat počet shluků k, do kterých budou vstupní data rozřazena:

1. Inicializace shluků – v této části hrají velkou roli náhodná čísla, díky kterým mohou různé inicializace nad stejnými daty vést k různým výsledkům.

   Existují dvě možnosti, jak inicializaci provést:

   - náhodně určit středy shluků
   - umístit každý objekt náhodně do shluku (v tomto případě se pokračuje krokem 3).

2. Umísti každý objekt do shluku, jehož střed je nejblíže.

3. Vypočítej nové středy shluků.

4. Opakuj kroky 2 a 3 dokud se nedosáhne konvergence nebo není splněna ukončující podmínka.

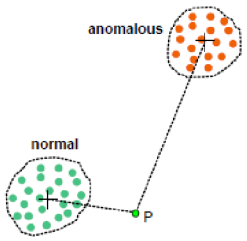
Na rozdíl od hierachického shlukování je časová složitost k-means lineární. Pro T iterací a K shluků nad M objekty charakterizovanými N atributy je složitost $O(T * K * M * N)$. Lineární složitost je jedním z důvodů, proč je tento způsob shlukování tolik oblíbený a proč si dokáže poradit i s velkým množstvím vstupních dat. Paměťová náročnost je $O(K + M)$.

Kvůli způsobu inicializace je výsledek algoritmu závislý na počáteční hodnotě (*seed*) generátoru náhodných čísel a oproti hierarchickému přístupu produkuje pouze hypersférické shluky.

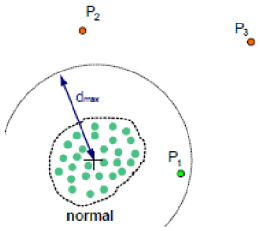
Anomální hosty je možné opět detekovat na základě počtu objektů v daném shluku – nižší počet hostů ve shluku než zadaný prah znamená označení všech těchto hostů jako potencionální útočníky.

4.2 Detekce s využitím trénovací fáze

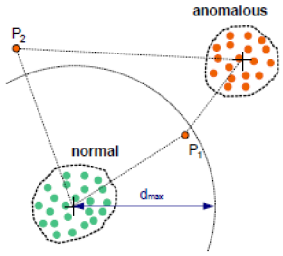
V předchozí sekci bylo popsáno, jak je možné rozeznat anomální provoz od normálního na základě počtu hostů ve shluku. To, že není potřeba trénovacích dat a trénovací fáze, je velkou výhodou shlukovacích algoritmů v kontextu detekce anomálií. Přesto existují přístupy, které využívají trénovací fáze a lze tak anomálie odhalit i jinými způsoby. Principy budou vysvětleny podle [8] s využitím k-means, použité ilustrační obrázky jsou součástí [8].



Obrázek 4.2: Detekce pomocí normálních i anomálních shluků.



Obrázek 4.3: Detekce pomocí normálních shluků.



Obrázek 4.4: Kombinace obou přístupů.

Základem jsou vstupní trénovací data, podle kterých je vytvořeno k shluků pro normální a anomální síťový provoz. Rozpoznání shluku s anomálním provozem může být dosaženo buď manuálně nebo pomocí heuristik. Středy těchto shluků jsou následně použity pro klasifikaci nových hostů a rozhodování zda jde o anomálie. K detekci lze využít klasifikaci nebo detekci *outliers*.

Při klasifikaci rozhoduje menší vzdálenost k danému shluku. Pozorovaný host je považován za neškodného, pokud leží blíže středu shluku s normálním provozem a považován za anomálního, pokud leží blíže shluku definující anomální provoz. Tento způsob umožňuje rozpoznat známé druhy anomálií, protože detekovaný host má podobnou charakteristiku síťového provozu jako ostatní hosté ze shluku anomálií.

Při detekci *outliers* se využívá z trénovacích dat pouze shluků s normálním síťovým provozem. Pro detekci *outliers* postačí hodnota prahu určující maximální vzdálenost od středu shluku d_{max}, při jejímž překročení je objekt považovaný za anomálii. Protože detekce závisí pouze na shlucích definujících normální chování, lze spolehlivě detekovat i neznámé anomálie, které by v případě klasifikace nemusely být součástí trénovacích dat.

Použitím klasifikace i detekce *outliers* zároveň je možné využít klady obou přístupů a vylepšit tak samotnou detekci. Objekt je pak chápán jako anomálie, pokud leží blíže středu shluku s anomálním provozem nebo pokud je jeho vzdálenost od centra normálního shluku větší než d_{max}.

4.3 Network Data Mining

Detekce anomálií není jen o výběru algoritmu, který samotnou detekci zajistí. Je to celý proces začínající získáním dat o provozu v síti, jejich zpracování a transformace a končící vyhodnocením výsledků detekčního algoritmu. *Network data mining* je založen na klasickém *data miningu*, konkrétně na *Knowledge Discovery in Database*, s tím, že tyto principy aplikuje na velká množství dat popisující síťový provoz. Celý proces detekce anomálií pak může být rozdělen na následující podčásti [8]:

1. **Výběr nezpracovaných prvotních dat** – Vstupní data jsou běžně získávána přímo ze síťových zařízení, která monitorují provoz na síti, například v podobě Cisco NetFlow záznamů (viz. kapitola 2). Protože síťový provoz je generován neustále a ve velkém množství, je nutné získané záznamy rozdělit do menších časových oken, která se budou zpracovávat jednotlivě. V případě této práce se bude pracovat s pětiminutovými úseky.

2. **Preprocesing a transformace dat** – Z dat získaných v prvním kroku je nutné vyextrahovat použitelné informace, na základě kterých bude možné rozeznat anomálie od normálního provozu. Jednou z možností je vytvoření statistik o hostech – pro každou IP adresu je vytvořena statistika vycházející z NetFlow dat. Veškerá aktivita, ve které daná IP adresa figuruje, je tak sjednocena do jednoho záznamu.

   Dále je také možné v této fázi aplikovat různé filtry a vzorkování. Filtry mohou například vyřadit hosty, kteří nemají pro detekci anomálií velký vliv a význam. Vzorkování je někdy nutné použít v závislosti na použitém algoritmu pro detekci a hardwarových podmínkách. To může vést k nepřesným a horším výsledkům, ale bez vzorkování by nebylo možné detekci provést.

   Součástí transformace je výběr konkrétních atributů ze statistik hostů, které budou použity pro detekci. Použitý vektor vlastností (*feature vector*) tedy obsahuje jen nejdůležitější atributy nebo hodnoty odvozené z více atributů (např. poměry). Před spuštěním hlavního algoritmu pro detekci je také třeba normalizovat použité hodnoty.

3. **Data mining** – V této fázi je použit vybraný algoritmus dolování dat. V případě této práce je to buď aglomerativní shlukování, nebo k-means.

4. **Interpretace výsledků** – Nakonec dochází k interpretaci výsledků předchozího kroku, kdy se podle zadaných prahů rozhoduje, kteří hosté jsou anomální.

4.3.1 Výběr vektoru vlastností

Každý objekt je charakterizován vlastním n-rozměrným vektorem vlastností, který je použit k výpočtu podobností s ostatními objekty. Typ a počet zvolených vlastností ovlivňuje detekční schopnosti i efektivitu algoritmů. Položky vektoru mohou být přímo hodnoty získatelné z NetFlow statistik nebo mohou být odvozeny z více hodnot.

Pro hodnoty vektorů lze například využít statistiky o síťových tocích nebo statistiky o jednotlivých IP adresách (síťových hostech). V této práci se pracuje se statistikami hostů, které jsou shromážděny z NetFlow záznamů zachytávající pět minut síťového provozu. Všechny dostupné statistiky použitelné pro vektory vlastností jsou:

- Počet vstupních a výstupních paketů, bajtů a toků.
- Počty vstupních a výstupních TCP příznaků, které se během komunikace vyskytly – SYN, ACK, FIN, RST, PSH, URG.
- Počty unikátních IP adres, se kterými daný host komunikoval (cílové a zdrojové adresy).
- Čísla linek, na kterých byla zachycena příchozí/odchozí komunikace.

Použití všech 22 dostupných statistik může znatelně zpomalit výpočty shlukovacích algoritmů a navíc nemusí platit, že čím více statistik použijeme, tím dosáhneme lepších výsledků při detekci anomálií. Proto se obecně volí jen určitá podmnožina statistik, která zajistí rychlou a zároveň spolehlivou detekci.

Jeden z vektorů použitý v této práci vychází z osmi přímo dostupných hodnot ze statistik o hostech: příchozí/odchozí pakety, příchozí/odchozí bajty, příchozí/odchozí síťové toky a počet unikátních zdrojových a cílových IP adres, se kterými daný host komunikoval.

Druhý použitý vektor vychází z [3] a jeho hodnoty jsou ze statistik o hostech vypočítány. Celkem 12-ti rozměrný vektor vlastností je zobrazen v tabulce 4.1.

Příchozí	Odchozí	Poměr odchozí/příchozí
Bajty / Pakety	Bajty / Pakety	Bajty
Pakety / Toky	Pakety / Toky	Pakety
Toky / # Unikátní zdrojové IP	Toky / # Unikátní cílové IP	Toky
# Unikátní zdrojové IP	# Unikátní cílové IP	IP adresy

Tabulka 4.1: Vektor vlastností (poměry)

4.3.2 Normalizace dat

Dimenze obsažené ve vektoru vlastností spolu nesouvisí a pokrývají různé rozsahy, proto je nutná jejich normalizace, jejímž cílem je sjednotit rozsahy a zaručit tak, aby každá dimenze ovlivňovala hodnoty podobnosti stejnou měrou.

Použité techniky normalizace v této práci budou popsány pro vektory vlastností M hostů v síti – $X = \{x_1, x_2, ..., x_M\}$, kde $X(i)$ značí i-tou dimenzi vektorů $X(i) = \{x_1(i), x_2(i), ..., x_M(i)\}$.

Normalizace minimální a maximální hodnotou

Základní normalizace, která rozsah vstupních dat převede do jednotkového $< 0, 1 >$.

$$X(i) = \frac{X(i) - Min(X(i))}{Max(X(i)) - Min(X(i))}$$

Normalizace podle percentilů

Normalizace velmi podobná předchozí podle minima a maxima s tím rozdílem, že se při určování hraničních hodnot ignoruje prvních $x\%$ a posledních $y\%$ hodnot – normalizace se tedy počítá podle x-tého a y-tého percentilu. Zvolením spodní hranice percentilu na 10 a horní hranice na 90, lze normalizovat podle [3]:

$$X(i) = \frac{X(i)}{p_{90}(X(i)) - p_{10}(X(i))}$$

kde p_x je funkce vracející x-tý percentil.

Díky této normalizaci je 80% hodnot součástí jednotkového rozsahu, zatímco zbytek hodnot vyčnívá a má větší šanci způsobit, že bude daný host označen jako anomálie.

Normalizace pomocí průměru a směrodatné odchylky

Tento způsob bere na vědomí charakter vstupních dat. Využitím průměru ($Mean$) a směrodatné odchylky (Std) lze docílit podobného rozsahu napříč všemi dimenzemi vektoru vlastností [13].

$$X(i) = \frac{X(i) - Mean(X(i))}{Std(X(i))}$$

Normalizace s využitím logaritmů a odmocnění

Aplikování logaritmu nebo odmocnění na vstupní data zachová rozdíly u nízkých hodnot, zatímco změna pro rozdíly u vysokých hodnot už není tak markantní [6]. Je například velký rozdíl, zda pozorovaný host komunikoval s 1 nebo 10 IP adresami, ale už není tolik důležitý rozdíl mezi 1001 a 1010 IP adresami. Pouze použitím logaritmu nebo odmocnění se ovšem nedocílí normalizace, proto je nutné takto transformovaná data normalizovat jednou z metod zmíněných výše.

$$X(i) = \ln(X(i) + 1)$$

$$X(i) = \sqrt{(X(i))}$$

Ve srovnání s přirozeným logaritmem je druhá odmocnina rychleji rostoucí funkcí.

Kapitola 5

Popis implementace

Program byl implementován v jazyce C++ pod operačním systémem GNU/Linux s využitím knihovny CLUTO [1], ze které jsou použity funkce pro hierarchické a k-means shlukování.

CLUTO je volně dostupná knihovna s uzavřeným zdrojovým kódem určená pro vzdělávací a výzkumné účely. Je implementovaná v jazyce C a podporuje nejrůznější operační systémy a architektury. Knihovna nabízí široké spektrum algortimů pro shlukování, výpočet podobností a mnoho dalšího.

V programu jsou implementovány výše zmíněné normalizační techniky. Program také podporuje oba vektory vlastností popsané v kapitole 4.3.1. Všechny techniky lze libovolně kombinovat zvolením příslušných parametrů.

Účelem programu je srovnání a vyhodnocení různých normalizací a parametrů s cílem nalézt spolehlivou kombinaci, která bude schopná kvalitně a efektivně detekovat anomálie a která by mohla být použita v realném nasazení (například ve firemních sítích).

5.1 Hierarchické shlukování

Pro hierarchické shlukování byla použita funkce `CLUTO_SA_Cluster`, která pro svoji funkčnost vyžaduje matici vzájemných podobností mezi hosty (pole o velikosti M^2, kde M značí počet hostů). Podobnosti jsou vypočítány jako obrácené hodnoty Euklidovské čtvercové vzdálenosti. Pro dosažení lepších výsledků a zrychlení výpočtů se výsledná vzdálenost neodmocňuje – vzdálenosti mezi jednotlivými hosty jsou vyšší, což dává anomáliím větší možnost vyniknout. Funkce vrací hotové řešení, kdy jsou všichni hosté součástí kořenového shluku. Je tedy nutné pomocí dat, která jsou po dokončení shlukování k dispozici, najít bod, kdy se mělo podle hodnoty α shlukování zastavit.

Protože je hierarchické shlukování paměťově i časově náročné, byl nastaven limit určující maximální počet hostů, který může být shlukován. Limit byl experimentálně stanoven na 13000 hostů (pro vyšší hodnoty již funkce z CLUTO knihovny nepracuje korektně) a při jeho překročení jsou vstupní data patřičně navzorkována.

5.2 K-means shlukování

K-means shlukování využívá funkci `CLUTO_VP_ClusterDirect`, jejímž vstupem je pole obsahující vektory vlastností všech shlukovaných hostů (velikost pole je tedy $M * N$, kde M

[1] Informace o CLUTO knihovně lze nalézt na *http://glaros.dtc.umn.edu/gkhome/views/cluto*. V práci byla použita verze 2.1.2a

značí počet hostů a N počet dimenzí vektoru). Výsledkem je pole, jehož indexy reprezentují jednotlivé hosty a uložené hodnoty značí, do jakého shluku byl host umístěn. Funkce navíc podporuje možnost vypočítat shlukování vícekrát a vybrat nejlepší řešení. Počet těchto pokusů je v programu pevně nastaven na 10.

5.3 Vstupní data

Program pracuje se soubory obsahující statistiky o IP adresách z 5-ti minutového časového úseku. Statistiky jsou vypočteny z NetFlow záznámů, ale jejich zpracování není součástí práce. Statistiky o hostech (*host stats*) poskytla výzkumná skupina ANT[2] z VUT FIT.

5.4 Překlad a spuštění

Překlad programu pod operačními systémy typu GNU/Linux je možný pomocí překladače **g++** a programu **make** (soubor `makefile` je součástí zdrojových kódů). Zdrojové kódy jsou přeloženy do jednoho spustitelného souboru pojmenovaného jako `bpClust`.

Při drobných změnách v programu, které se týkají výpisu IP adres do příkazové řádky, lze snadno docílit přenositelnosti kódu i na jiné operační systémy. Je ale nutné použít kompatibilní CLUTO knihovnu.

5.4.1 Vstupní parametry

`--input <soubor>`

Vstupní soubor se statistikami o hostech. Povinný parametr.

`--clust <k/h>`

Výběr shlukovacího algoritmu – k = k-means, h = hierarchické shlukování. Povinný parametr.

`--features <orig/ratios>`

Určení jaký vektor vlastností bude použit (orig = hodnoty přístupné přímo ze statistik, ratios = hodnoty vypočítané – viz. tabulka 4.1). Defaultně je nastavena volba `orig`.

`--normalization <minmax/percentile/meanstd>`

Výběr normalizační techniky. Defaultně není použita žádná, ale je povinné zadat alespoň jednu techniku pro normalizaci nebo prenormalizaci.

`--prenormalization <log/sqrt>`

Možnost výběru mezi předzpracováním dat před samotnou normalizací buďto pomocí přirozeného logaritmu nebo odmocněním. Defaultně není prenormalizace nastavena.

`-c <POCET SHLUKU>`

Důležitý parametr pro k-means algoritmus – určuje do kolika výsledných shluků budou data shlukována. Defaultně je nastavená hodnota 20.

[2] ANT – Accelerated Network Technologies

`-a <ALFA>`

Hodnota α určuje práh, který rozhoduje o zastavení hierarchického shlukování. Větší hodnota znamená větší pravděpodobnost, že nebude detekována žádná anomálie. Defaultně je nastavená hodnota 3.

`-l <POCET HOSTU>`

Značí kolik hostů ve výsledném shluku způsobí, že bude označen jako anomální a všichni hosté v něm tak budou označeni jako potencionální utočníci (jejich IP adresy budou po skončení programu vypsány). Defaultně nastaven limit na 5 hostů.

`-f <POCET PAKETU>`

Zapnutí jednoduchého filtrování hostů. Každý host, který zaslal a přijmul méně než zadaný počet paketů, nebude zahrnut do shlukovaní. Defaultně nastaveno na 0 – vypnuté filtrování.

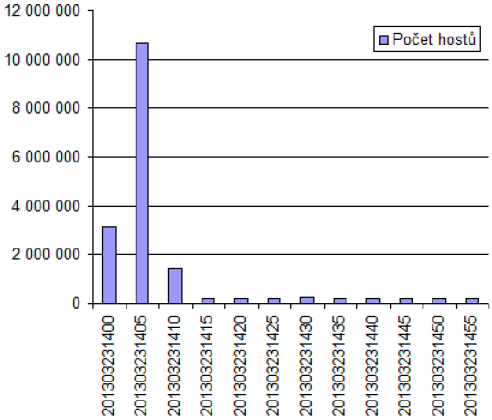
`--format simple/onlyip/indent/indentinfo`

Nastavení způsobu, jakým budou vypsány výsledky shlukování. Způsob `simple` vypíše na každý řádek číslo shluku a seznam IP adres, které do něj náleží. Způsob `onlyip` vypíše na každý řádek jednu IP adresu. Způsoby `indent` a `indentinfo` vypíší čísla shluků a seznam IP adres v čitelnější podobě (`indentinfo` navíc u každé IP adresy poskytne některé základní statistiky získané ze vstupního souboru). První dva způsoby výpisu jsou vhodné jako vstup do programů či skriptů pro další zpracování a porovnávání.

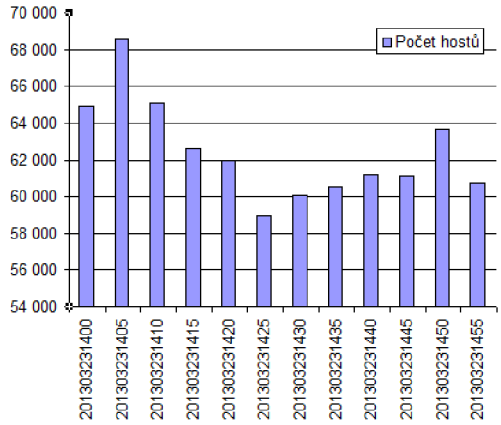
Kapitola 6

Vyhodnocení metod

Metody byly otestovány na anonymizovaných datech z univerzitní sítě VUT. Konkrétně se jedná o statistiky ze dne 23.3.2013 od 14:00 do 15:00. Celkem je tedy dostupných 12 souborů, každý obsahující statistiky z 5-ti minut síťového provozu. Pro prvotní seznámení a náhled na tato data lze použít jednoduchý graf zobrazující počty unikátních IP adres (hostů), které se během daných časových intervalů vyskytly.



Obrázek 6.1: Počty hostů vyskytující se během uvedených 5-ti minutových intervalů.



Obrázek 6.2: Počet hostů po aplikaci filtrování nastaveným na 5 paketů.

Z prvního grafu 6.1, který zobrazuje originální počty hostů v souborech se statistikami, je patrné, že v prvních 15 minutách došlo k značnému vychýlení od normálního chování v síti. Normální chování sítě by se na základě dostupných dat dalo definovat přibližně 200 tisíci hosty na 5 minut (přesto i zde se mohou vyskytovat anomálie), ale v prvních 15 minutách vystoupal počet unikátních hostů až za hranici 10-ti miliónů a lze s jistotou říct, že jde o nějaký typ útoku/anomálie.

Shlukování nad 10 milióny objekty (ale i méně) je vzhledem k časové i paměťové náročnosti použitých algoritmů nemyslitelné. Například hierarchické shlukování by pro uložení matice podobností pro 1 milión hostů vyžadovalo téměř 2TB dat, pro 200 tisíc pak 74GB. Pro dosažení reálnějších hodnot výpočetních časů a paměťových nároků je na vstupní statistiky použit filtr, který vyřadí hosty s méně než zadaným počtem příchozích a odchozích paketů. Ačkoliv tito hosté tvoří přibližně dvě třetiny z celkového počtu (filtrování pro 5

paketů), jsou z hlediska detekce anomálií méně důležití a neměli by výsledky detekce příliš ovlivnit. Počty hostů po vyfiltrování jsou zobrazeny ve 2. grafu 6.2. Filtrování způsobilo snížení počtu hostů na 60-70 tisíc za 5 minut síťového provozu. Už z této skutečnosti lze vyvodit, že pokud proběhl v prvních 15 minutách nějaký rozsáhlý útok, drtivá většina hostů byla pravděpodobně cílem tohoto útoku a neměla důvod zasílat či přijímat větší množství paketů.

6.1 Vybrané útoky a anomálie v počítačových sítích

Útoků v počítačových sítích existuje velké množství a neustále vznikají nové dosud neznámé varianty. Ne všechny útoky je však snadné detekovat, některé se mohou zcela ztratit v normálním provozu v síti. Zároveň neplatí, že všechny anomálie musí být útoky. Například i výpadek serveru může způsobit anomálie v síťovém provozu. V následující části kapitoly budou popsány některé typy útoků a anomálií, se kterými se při testování můžeme setkat [5].

- **Alfa tok** – Vyznačuje se neobyvkle vysokým tokem dat mezi dvěma body v síti. Lze ho charakterizovat vysokým počtem příchozích/odchozích paketů a bajtů a nízkým počtem toků a IP adres.
- **DoS a DDoS útoky** – Takzvané odepření služby (*Denial of Service*) je útok proti jednomu objektu v síti, který poskytuje často důležitou službu pro ostatní uživatele. Útok může mít jeden, ale i více zdrojů, pak se jedná o distribuovaný (*Distributed*) útok. U útočníků jsou ovlivněny počty paketů, bajtů a toků směřující na jedinou cílovou IP adresu (oběť). U některých typů DoS útoků mohou být ovlivněny i různé TCP příznaky.
- **Skenování IP adres** – Hledání aktivních IP adres často v adresovém prostoru určité organizace. K útoku se využívá převážně protokolu ICMP. Vyšší a zároveň stejné počty paketů, toků a cílových IP adres jsou typické pro tento útok.
- **Skenování portů** – Zjišťování otevřených portů u jedné nebo více cílových adres. Charakter útoku je podobný jako u skenování IP adres.
- **Flash Crowd** – Velký nárust komunikace s určitým objektem v síti. Může jít o zájem o novou službu. Charakter je podobný jako u DDoS útoků, ale tato anomálie většinou trvá kratší dobu.
- **Jeden na mnoho** – Distribuce obsahu z jednoho serveru k více uživatelům. Jedná se spíše o anomálii charakterizovanou vysokým počtem paketů a bajtů odeslaných z jedné zdrojové adresy na více cílových.
- **Výpadek** – Výrazný pokles provozu, který může značit selhání či údržbu serverů.

6.2 Způsob vyhodnocení

Testování všech možných kombinací, které program umožňuje, by bylo velmi náročné a poskytnuté výsledky by byly rozsáhlé a do značné míry matoucí. Na základě prvotního experimentování s různými parametry byly proto vybrány pouze kombinace, poskytující zajímavé a přesnější výsledky.

Z normalizačních technik byly pro účely testování vyřazeny techniky využívající minimum-maximum a logaritmování. Normalizace pomocí minima-maxima se ve srovnání s ostatními normalizačními technikami nehodí pro účely detekce anomálií, protože lineárně převádí vstupní hodnoty do pevně daného cílového intervalu. Při výskytu výrazné anomálie tak může způsobit utlumení hodnot pro ostatní méně výrazné anomálie, které pak budou klasifikovány jako běžný provoz. Převod hodnot s využitím logaritmu byl vyřazen, protože ve srovnání s odmocňováním nedává tolik prostoru pro vyšší hodnoty. Například anomálie s vysokým počtem toků mají při použití logaritmování větší šanci sloučit se do jednoho shluku a tím tak znemožnit detekci (pokud vytvoří dostatečně velký shluk).

Použité zkratky při zpracování výsledků

Ř – číslo řádku

P – normalizace pomocí percentilů.

M – normalizace pomocí průměru a směrodatné odchylky.

S – předzpracování dat pomocí odmocnění.

V1 – použití vektoru vlastností, který obsahuje příchozí/odchozí pakety, bajty, toky a IP adresy.

V2 – použití druhého vektoru obsahujícího poměry různých statistik (viz. tabulka 4.1).

Například **V2-SP** značí, že byl použit vektor vlastností 2, pro předzpracování dat odmocnění a pro normalizaci percentily.

Čísla uvedená ve sloupcích pod jednotlivými kombinacemi značí počet detekovaných anomálií, konkrétně jde o dvě čísla: *počet detekovaných anomálií algortimem/z toho skutečných anomálií*. Nejlepší výsledky detekce jsou ty, kde je rozdíl mezi těmito počty minimální a zároveň je dosaženo vysokého počtu skutečných anomálií. Počet skutečných anomálií byl manuálně ověřen ve spolupráci s vedoucím bakalářské práce a to na základě originálních NetFlow záznamů. Jako skutečné anomálie byly označeni hosté, u kterých bylo zřejmé, že jde o určitý druh útoku, nebo u kterých se nepodařil nijak vysvětlit generovaný provoz, ale byl natolik neobvyklý, že jej bylo vhodné označit také.

Testování proběhlo na studentském školním serveru *ant-stud* – Intel Xeon 2.5GhZ (4 jádra), 24GB RAM, GNU/Linux 64-bit.

6.3 Hierarchické shlukování

K ověření detekce hierarchického shlukování jsou použity statistiky o hostech vypočítané z provozu jen na vybraných portech. Při shlukování nad souborem statistik z veškerého provozu by muselo být kvůli výpočetní náročnosti metody použito výrazné vzorkování, které by vedlo ke snížení kvality detekce. Navíc bylo pro urychlení výpočtů použito filtrování dat pomocí parametru `-f POCET PAKETU` (nastaveno na 10 paketů). Počet hostů, který tvoří hranici mezi anomálními a normálními shluky, je standardně nastaven na 5.

TCP port 22 (SSH)

SSH (*Secure Shell*) je komunikační protokol sloužící k bezpečnému a šifrovanému připojení na vzdálený počítač. Protokol nahradil méně zabezpečené protokoly Telnet, Rlogin a Rsh. Pomocí SSH lze ovládat vzdálený terminál, přenášet data a další. Při nevhodné konfiguraci (například neomezení počtu pokusů k přihlášení), lze narazit na útoky hrubou silou, které mají za cíl prolomit přístupové heslo.

V tabulce 6.1 jsou zobrazeny počty detekovaných anomálií pro vybrané časové úseky.

V čase 14:05 – 14:10 byl detekován host s IP adresou 76.26.22.229[1], který způsobil velký nárust počtu komunikujících hostů ("špička"viditelná v grafu 6.1). Podle generovaného provozu šlo o skenování sítě pomocí SYN paketů s cílem nalézt otevřené SSH porty. Dvě další detekované anomálie se vyznačovaly vysokým přenosem dat – alfa tok. Poslední detekovaná anomálie generovala podezřelý provoz. V úseku 14:10 – 14:15 pokračovala adresa 76.26.22.229 ve skenování, stejně tak byl detekován i host generující podezřelý provoz. Nově bylo detekováno skenování z adresy 127.212.150.173 (detekováno také v 14:20 – 14:25). V čase 14:20 – 14:25 se IP adresa 76.26.22.229 začala připojovat na SSH na různé adresy (pravděpodobně získané z předešlého skenování). V posledním časovém intervalu bylo nově detekováno další větší skenování pomocí SYN paketů. Ostatní detekováné adresy se objevily už dříve.

Chybné detekce (falešné poplachy) byly často způsobeny adresami přenášejícími větší množství dat, které se odlišovalo od množství datových přenosů ostatních adres, ale i přesto byl tento provoz zcela legitimní. Nicméně detekci, která proběhla v pořadí nad přibližně 150/2500/4500/100 hosty, lze v tomto případě označit za velmi úspěšnou. Detekované adresy se napříč jednotlivými kombinacemi nelišily (pouze jejich počet). Podle získaných výsledků se nejlepší detekce dosahovalo s hodnotou parametru $alfa = 3$, která na rozdíl od $alfa = 2$ generovala minimální množství falešných poplachů (nejčastěji 0-1). Zajímavý je také počet falešných poplachů pro jednotlivé vektory vlastností (zvlášť viditelné rozdíly jsou pro $alfa = 2$) – v prvním časovém úseku, kdy se vyskytla "špička", použití vektoru `V2` vygenerovalo větší počet falešných poplachů než `V1`, pro ostatní časy ovšem `V2` generovalo menší počet chybných detekcí než `V1`. Kombinace `V2-M` se ukázala být nejméně úspěšnou, ostatní kombinace generovaly srovnatelné výsledky.

TCP port 25 (SMTP)

Protokol SMTP (*Simple Mail Transfer Protocol*) je určený pro přenos zpráv elektronické pošty. Systém doručování stojí na třech druzích programů: poštovní klient, poštovní server a program pro lokální doručování zpráv. Při detekci anomálií na tomto portu je možné narazit například na rozesílání většího množství zpráv více příjemcům, tedy rozesílání spamu.

S očekáváním detekce rozesílání spamu nebyla potvrzena během hodinového provozu žádná anomálie (tabulka 6.2). Detekovány byly různé poštovní servery a uživatelé zasílající objemnější přílohy. Chybná detekce mohla být zapříčiněna nízkým počtem hostů (průměrně 250), nad kterým shlukování proběhlo, a obecně nižším využitím služby (například ve srovnání s SSH) – vyšší počet poštovních serverů a klientů zasílající větší přílohy by mohl vést k nižšímu počtu falešných detekcí.

Celkově shlukování generovalo velké množství falešných poplachů, které by byly pravděpodobně vygenerovány i v případě, kdy by se v datech vyskytovala skutečná anomálie. Lepší detekce se dosahovalo pro hodnoty $alfa = 5$ a více.

[1] Všechny IP adresy v této práci byly pozměněny a neodpovídají skutečným adresám.

Ř	Čas. úsek	Alfa	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:05 – 14:10	2	5/4	17/3	16/4	22/4	37/4	2/2
2	14:05 – 14:10	3	5/4	4/3	3/3	5/4	2/2	2/2
3	14:05 – 14:10	5	5/4	4/3	3/3	2/2	2/2	2/2
4	14:05 – 14:10	10	3/3	4/3	3/3	2/2	0	2/2
5	14:10 – 14:15	2	32/2	23/2	3/2	6/2	7/2	5/2
6	14:10 – 14:15	3	3/2	3/1	3/2	4/2	3/2	5/2
7	14:10 – 14:15	10	3/2	1/1	3/2	4/2	0	2/1
8	14:20 – 14:25	2	17/3	11/3	9/3	11/3	7/3	1/1
9	14:20 – 14:25	5	4/2	2/1	2/1	2/2	0	1/1
10	14:30 – 14:35	2	7/3	20/4	5/3	7/4	14/4	7/4
11	14:30 – 14:35	3	3/3	20/4	1/1	7/4	2/2	5/4
12	14:30 – 14:35	5	3/3	1/1	1/1	7/4	0	3/2
13	14:30 – 14:35	10	1/1	1/1	1/1	3/2	0	1/1

Tabulka 6.1: Počet nalezených anomálií pro TCP port 22

Ř	Čas. úsek	Alfa	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:00 – 14:05	2	18/0	6/0	5/0	12/0	3/0	3/0
2	14:00 – 14:05	3	18/0	6/0	0	12/0	0	3/0
3	14:00 – 14:05	5	5/0	6/0	0	2/0	0	0
4	14:15 – 14:20	2	13/0	9/0	6/0	5/0	10/0	3/0
5	14:15 – 14:20	3	6/0	6/0	6/0	3/0	0	3/0
6	14:15 – 14:20	4	6/0	3/0	6/0	3/0	0	0
7	14:15 – 14:20	10	6/0	0	0	0	0	0
8	14:45 - 14:50	2	9/0	4/0	4/0	2/0	2/0	1/0
9	14:45 - 14:50	3	3/0	3/0	4/0	2/0	2/0	0
10	14:45 - 14:50	5	3/0	3/0	0	2/0	0	0

Tabulka 6.2: Počet nalezených anomálií pro TCP port 25

TCP port 80 (HTTP)

HTTP (*HyperText Transfer Protocol*) slouží pro přenos hypertextových dokumentů a jeho vznik je spojen s WWW (*World Wide Web*). Klienti (webové prohlížeče) pomocí protokolu HTTP stahují a zobrazují dokumenty uložené na serverech a generují tak velkou část síťového provozu. Protokol je nezabezpečený, což dává možnost k odposlechu komunikace. Zabezpečená verze HTTPS komunikuje na portu 443. Časté jsou útoky na servery s využitím různých typů DoS/DDoS.

Během testování nebyla detekována žádná extrémní anomálie, ale přesto provoz některých IP adres působil dostatečně podezřele na to, aby byl označen jako anomální. Falešné pozitivní detekce byly způsobeny vytíženějšími webovými servery a klienty přenášející vyjímečné množství dat. Vzhledem k počtu hostů, který přesahoval 8000, a podobnému charakteru generovaného provozu se většina hostů sloučila už při nízkých hodnotách alfa. Výsledky detekce nad portem 80 shrnuje tabulka 6.3. Použití normalizace pomocí průměru a směrodatné odchylky (`V1-M` a `V2-M`) detekovalo nejvíce skutečných anomálií.

Ř	Čas. úsek	Alfa	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:10 – 14:15	2	7/2	14/5	2/0	4/1	7/3	3/1
2	14:10 – 14:15	3	7/2	8/3	0	4/1	7/3	3/1
3	14:10 – 14:15	5	0	0	0	3/1	0	0
4	14:30 – 14:35	2	4/0	11/4	4/0	9/3	11/5	2/2
5	14:30 – 14:35	5	4/0	0	0	2/2	0	2/2
6	14:40 – 14:45	2	4/0	10/4	2/0	10/2	1/0	1/0
7	14:40 – 14:45	5	4/0	0	2/0	1/0	0	1/0

Tabulka 6.3: Počet nalezených anomálií pro TCP port 80

UDP port 53 (DNS)

DNS (*Domain Name System*) servery tvoří hierarchickou distribuovanou síť, určenou k uchování a poskytování informací o jednotlivých doménových jménech. Jednou z nejdůležitějších služeb DNS je překlad doménových jmen na IP adresy. Podobně jako na webové servery lze i na DNS servery snadno uskutečnit DoS/DDoS útok, nicméně typů útoků cílených na DNS existuje mnoho.

Výsledky detekce jsou zobrazeny v tabulce 6.4. Podobně jako u SSH byl celkově vygenerován menší počet falešných poplachů. Detekované IP adresy, jejichž aktivita byla ověřena a označena za anomální, se vyskytují ve všech testovaných časových úsecích.

V čase 14:15 – 14:20 byl detekován s velkou pravděpodobností typ DoS útoků, kdy oběť (76.26.12.200) i útočníci (129.195.178.118, 213.210.100.247) byly úspěšně označeny jako anomálie. Tito útočníci byly kromě kombinace `V2-M` úspěšně detekováni všemi kombinacemi pro hodnotu $alfa = 2$. Další IP adresy, které mohly být součástí zmíněného DoS útoku, byly detekovány v pozdějších časech. Několik detekovaných adres bylo po kontrole označených jako DNS servery. Menší část těchto DNS serverů byla označena jako anomální, protože byl na tyto servery z určitých adres generován nadměrný (anomální) provoz (DDoS). Zbylé DNS servery byly detekovány chybně a tvořily tak většinu falešných poplachů.

Při detekci se osvědčila kombinace `V1-P`, která dosahovala nejlepších výsledků, kdy i pro nízké hodnoty `alfa` ($alfa = 2$) detekovala větší počet anomálií než falešných poplachů a než počet anomálií, které detekovaly ostatní kombinace s vyšší hodnotou `alfa`.

Protokol ICMP

ICMP (*Internet Control Message Protocol*) je protokol síťové vrstvy určený především k diagnostice a kontrole spojení mezi body v síti. Slouží také jako prostředek pro zasílání základních chybových zpráv. ICMP lze snadno využít k rozsáhlému a rychlému skenování sítě (pomocí zpráv *Echo request*).

V čase 14:00 – 14:05 bylo odhaleno skenování sítě ze dvou IP adres. Rozsáhlejší skenování bylo úspěšně odhaleno všemi kombinacemi a bylo detekováno ve všech testovaných časových úsecích. Druhá výrazná anomálie (76.26.22.229), je spojena se skenováním uskutečněným na portu 22 (detekována i v čase 14:10 – 14:15). Za anomální byly označeny i dvě IP adresy, které pravděpodobně testovaly stav VUT sítě – skenování sítě.

Pro nízké hodnoty alfa (tabulka 6.5) je v případě ICMP generován velký počet falešně pozitivních detekcí. To je způsobeno nízkým počtem anomálií (odhaleno bylo méně než 10 anomálií z celkového počtu 6000+ hostů), jejichž provoz se navíc diametrálně liší od normálního. Pro úspěšnější detekci je tedy třeba použít vyšší hodnoty alfa.

Ř	Čas. úsek	Alfa	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:15 – 14:20	2	18/10	5/4	4/4	27/6	2/2	23/5
2	14:15 – 14:20	3	4/4	2/2	4/4	24/6	2/2	1/1
3	14:15 – 14:20	5	4/4	2/2	0	8/5	0	1/1
4	14:30 – 14:35	2	18/12	2/2	2/2	25/8	3/3	17/7
5	14:30 – 14:35	3	4/4	2/2	2/2	20/8	3/3	1/1
6	14:30 – 14:35	5	4/4	2/2	2/2	17/7	3/3	1/1
7	14:30 – 14:35	10	1/1	2/2	0	1/1	0	1/1
8	14:40 – 14:45	2	18/12	20/12	18/12	31/8	3/3	10/5
9	14:40 – 14:45	3	18/12	2/2	18/12	8/5	0	10/5
10	14:40 – 14:45	7	1/1	2/2	0	8/5	0	0
11	14:50 – 14:55	2	20/10	18/10	18/10	22/6	7/4	16/5
12	14:50 – 14:55	3	6/4	2/2	2/2	22/6	4/3	16/5
13	14:50 – 14:55	10	1/1	2/2	0	1/1	0	0

Tabulka 6.4: Počet nalezených anomálií pro UDP port 53

Ř	Čas. úsek	Alfa	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:00 – 14:05	2	1200/6	1200/6	6/6	1207/6	1207/6	1207/6
2	14:00 – 14:05	3	1200/6	3/3	6/6	3/3	1207/6	1207/6
3	14:00 – 14:05	4	1200/6	3/3	2/2	3/3	0	2/2
4	14:00 – 14:05	5	6/6	3/3	2/2	3/3	0	0
5	14:10 – 14:15	3	1059/4	1059/4	1065/4	1065/4	6/2	1065/4
6	14:10 – 14:15	4	1059/4	1059/4	1065/4	1065/4	6/2	1065/4
7	14:10 – 14:15	5	4/4	1059/4	1065/4	1065/4	6/2	1065/4
8	14:10 – 14:15	15	4/4	4/4	4/4	2/2	0	0
9	14:35 – 14:40	4	3/3	868/4	3/3	3/3	868/4	6/4
10	14:35 – 14:40	5	3/3	3/3	3/3	3/3	4/2	6/4

Tabulka 6.5: Počet nalezených anomálií pro protokol ICMP

6.4 K-means shlukování

Pro testování k-means shlukování byla použita stejná data jako při hierarchickém shlukování i stejné parametry. K-means shlukování bylo použito i na vybrané časové úseky obsahující veškerý síťový provoz.

TCP port 22 (SSH)

Pro port 22 byly detekovány stejné anomálie jako u hierarchického shlukování. Kvalitní detekce je dosaženo až pro velmi nízké počty shluků (dáno celkovým počtem hostů). Zajímavá je detekce pro 2 shluky, kdy jsou spolehlivě detekovány nejvýraznější anomálie (76.26.22.229). Výsledky jsou shrnuty v tabulce 6.6.

TCP port 25 (SMTP)

Detekce na portu 25 byla stejně neúspěšná jako u hierarchického shlukování, viz. tabulka 6.7. Pro další testování nad tímto portem lze doporučit nízké počty shluků (5 a méně).

Ř	Čas. úsek	Shluků	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:05 – 14:10	20	38/4	37/3	40/3	40/4	27/3	34/4
2	14:05 – 14:10	10	14/4	17/3	14/4	15/4	20/3	12/3
3	14:05 – 14:10	5	6/3	7/3	6/3	8/3	2/2	4/2
4	14:05 – 14:10	2	2/1	2/2	2/1	2/1	0	2/1
5	14:10 – 14:15	30	16/2	54/1	7/1	11/2	9/2	6/1
6	14:10 – 14:15	20	13/2	36/1	7/1	6/1	6/2	2/1
7	14:10 – 14:15	10	13/2	16/1	7/1	6/2	4/2	2/1
8	14:10 – 14:15	5	9/2	6/1	2/1	4/1	3/1	2/1
9	14:20 – 14:25	20	14/3	37/1	6/1	2/1	7/3	0
10	14:20 – 14:25	10	14/3	16/1	2/1	2/1	7/1	0
11	14:20 – 14:25	5	4/2	6/1	2/1	2/1	5/0	0
12	14:30 – 14:35	20	34/4	42/4	45/4	28/5	40/4	32/5
13	14:30 – 14:35	10	18/3	18/1	16/3	13/4	15/3	19/4
14	14:30 – 14:35	5	12/3	5/1	2/1	8/3	6/3	4/3
15	14:30 – 14:35	2	2/1	0	2/2	2/1	0	2/2

Tabulka 6.6: Počet nalezených anomálií pro TCP port 22

Ř	Čas. úsek	Shluků	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:00 – 14:05	10	8/0	14/0	4/0	6/0	8/0	3/0
2	14:00 – 14:05	5	6/0	6/0	4/0	5/0	4/0	3/0
3	14:00 – 14:05	2	2/0	0	0	2/0	0	3/0
4	14:15 – 14:20	20	24/0	34/0	17/0	25/0	12/0	16/0
5	14:15 – 14:20	10	6/0	16/0	6/0	9/0	7/0	2/0
6	14:15 – 14:20	5	6/0	6/0	4/0	7/0	0	3/0
7	14:15 – 14:20	2	2/0	0	0	2/0	0	3/0
8	14:45 – 14:50	10	10/0	14/0	2/0	12/0	9/0	7/0
9	14:45 – 14:50	5	6/0	5/0	5/0	5/0	8/0	0
10	14:45 – 14:50	2	3/0	0	0	3/0	0	0

Tabulka 6.7: Počet nalezených anomálií pro TCP port 25

TCP port 80 (HTTP)

Pro detekci nad portem 80 byl navýšen limit hostů na 10, což vedlo ke zvýšení počtu detekovaných anomálií (ve srovnání s hierarchickým shlukováním) – tabulka 6.8. Předpokladem bylo, že se tak dosáhne i vyššího počtu pravých pozitivních detekcí. Nové anomálie ale nakonec detekovány nebyly.

Vzhledem k vyššímu počtu hostů, nad kterými je shlukování spouštěno, je pro získání kvalitnějších výsledků nutné zvýšit počet cílových shluků. Použití kombinací `V1-SP` a `V2-SP` vedlo k méně úspěšné detekci.

UDP port 53 (DNS)

Pro první dva časové úseky (14:15 – 14:20 a 14:30 – 14:35) byl podobně jako u detekce na portu 80 zvýšen limit hostů na 10. Ani v tomto případě se ale nepotvrdily žádné nové

Ř	Čas. úsek	Shluků	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:10 – 14:15	30	30/4	36/5	7/0	12/3	26/3	3/1
2	14:10 – 14:15	20	28/4	31/4	0	23/3	30/2	3/1
3	14:10 – 14:15	10	7/1	18/3	0	7/3	16/2	3/1
4	14:10 – 14:15	5	0	0	0	4/1	0	3/1
5	14:30 – 14:35	30	22/0	32/1	8/0	19/7	12/5	6/3
6	14:30 – 14:35	20	7/1	21/3	0	13/5	20/5	12/3
7	14:30 – 14:35	10	4/0	12/3	0	13/5	6/4	2/2
8	14:40 – 14:45	30	20/0	43/3	6/0	13/3	28/2	2/0
9	14:40 – 14:45	20	6/2	26/4	0	13/2	9/3	2/0
10	14:40 – 14:45	10	4/0	11/4	0	5/2	5/2	2/0

Tabulka 6.8: Počet nalezených anomálií pro TCP port 80

anomálie, než které byly detekovány hierarchickým shlukováním. Výsledky detekce jsou zobrazeny v tabulce 6.9.

Podobně jako u hierarchického shlukování se v tomto případě osvědčila normalizace pomocí percentilů.

Ř	Čas. úsek	Shluků	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:15 – 14:20	30	45/11	60/5	17/8	41/7	20/5	30/5
2	14:15 – 14:20	20	33/7	42/5	9/6	56/7	19/5	16/5
3	14:15 – 14:20	10	23/11	22/5	4/3	17/6	16/3	13/5
4	14:15 – 14:20	5	16/7	4/4	4/4	22/5	4/4	13/5
5	14:30 – 14:35	20	36/12	49/5	6/6	72/11	17/9	26/7
6	14:30 – 14:35	10	32/12	22/7	6/6	20/8	14/7	18/7
7	14:30 – 14:35	5	17/13	12/1	6/6	9/7	0	16/7
8	14:40 – 14:45	20	26/8	32/3	8/3	19/5	10/4	0
9	14:40 – 14:45	10	6/6	12/3	3/3	8/5	2/1	0
10	14:40 – 14:45	5	6/6	4/1	5/5	8/5	0	0
11	14:50 - 14:55	20	14/7	30/2	5/3	40/6	12/4	0
12	14:50 - 14:55	10	6/4	12/2	0	11/1	2/1	0
13	14:50 - 14:55	5	6/4	4/1	0	3/1	0	0

Tabulka 6.9: Počet nalezených anomálií pro UDP port 53

Protokol ICMP

K-means pro ICMP protokol detekoval shodné IP adresy jako hierarchické shlukování a díky pevně nastavenému počtu cílových shluků nebylo vygenerováno nadměrné množství falešných detekcí (tabulka 6.10).

Kvalitní detekce se dosahovalo při počtu shluků 10 a méně.

Veškerý provoz

Výsledky detekce nad veškerým provozem jsou zobrazeny v tabulce 6.11. Bylo detekováno několik desítek anomálií, u kterých nebylo možné ani na základě originálních NetFlow zá-

Ř	Čas. úsek	Shluků	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:00 – 14:05	30	29/6	55/2	2/2	10/6	7/3	7/2
2	14:00 – 14:05	20	7/6	37/4	2/2	10/6	7/3	2/2
3	14:00 – 14:05	10	7/6	14/3	2/2	10/6	5/3	2/2
4	14:00 – 14:05	5	7/6	9/2	2/2	7/6	0	2/2
5	14:10 – 14:15	30	10/4	55/2	5/2	8/4	10/2	6/2
6	14:10 – 14:15	20	22/4	32/2	2/2	8/4	5/2	6/2
7	14:10 – 14:15	10	7/4	16/2	4/2	8/4	5/2	6/2
8	14:10 – 14:15	5	6/4	4/2	4/4	5/4	0	2/2
9	14:35 – 14:40	20	2/2	32/2	3/3	6/3	6/3	6/4
10	14:35 – 14:40	10	2/2	14/2	3/3	5/3	5/3	3/3
11	14:35 – 14:40	5	2/2	8/2	0	2/2	5/2	6/3
12	14:35 – 14:40	2	2/2	0	0	2/2	0	0

Tabulka 6.10: Počet nalezených anomálií pro protokol ICMP

znamů určit, zda jde opravdu o škodlivý provoz. Tyto IP adresy nebyly zahrnuty do konečného počtu ověřených anomálií a je tedy možné, že skutečný počet pravých pozitivních detekcí je o něco vyšší.

V časovém úseku 14:05 – 14:10 byly detekovány IP adresy, které byly již dříve odhaleny na jednotlivých portech. Celkem bylo detekováno 5 IP adres (2x SSH, 2x HTTP a 1x DNS).

V druhém časovém úseku byla detekce některých kombinací úspěšnější – celkem 9 IP adres (6x HTTP, 2x SSH a 2x DNS).

Ř	Čas. úsek	Shluků	V1-P	V1-M	V1-SP	V2-P	V2-M	V2-SP
1	14:05 – 14:10	50	18/4	95/3	2/2	24/5	29/5	7/2
2	14:05 – 14:10	30	14/4	53/4	2/2	19/5	17/5	2/2
3	14:05 – 14:10	10	9/4	20/4	0	19/4	18/4	2/2
4	14:30 – 14:35	40	21/3	63/3	0	29/7	39/9	17/2
5	14:30 – 14:35	20	15/2	45/3	0	20/7	26/8	2/2
6	14:30 – 14:35	10	15/2	21/3	0	20/7	21/8	4/2

Tabulka 6.11: Počet nalezených anomálií pro veškerý síťový provoz

6.5 Shrnutí a porovnání metod

Obě metody byly schopné z testovacího vzorku dat pomocí vybraných kombinací detekovat všechny IP adresy, které byly manuálně ověřeny a označeny za anomální. Ze získaných výsledků není snadné určit jaká metoda je pro detekci vhodnější. K-means shlukování generovalo více falešně pozitivních detekcí (kromě detekce pro ICMP), ovšem při vhodně zvoleném počtu shluků se detekční schopnost vyrovnala hierarchickému shlukování. Velkou nevýhodou k-means je právě určení počtu shluků, nicméně existují funkce, které jsou schopné přibližně určit ideální počet shluků. Ideální počet shluků velmi závisí na celkovém počtu hostů i na charakteru síťového provozu. Hierarchické shlukování je v tomto ohledu mnohem více flexibilní. Přesto je třeba vhodně zvolit i hodnotu alfa. Nevýhoda hierarchického shlukování je jeho časová a paměťová náročnost.

Podobně jako u jednotlivých shlukovacích metod je obtížné vybrat i ideální typ normalizace a vektoru vlastností. Rozdíly v detekci při použití různých vektorů vlastností jsou ve většině případů nepatrné a navíc velmi závisí na aktuálním vzorku dat, se kterým se pracuje. Na základě detekce nad celým souborem statistik byl úspěšnější vektor vlastností s poměry. Z normalizačních technik se osvědčila normalizace pomocí percentilů (bez odmocnění), která dává možnost anomáliím skutečně vyniknout a tím zvýšit šanci k jejich detekci. V některých případech ale byla mnohem spolehlivější normalizace pomocí průměru a směrodatné odchylky. Jednoznačně tedy nelze vybrat ani normalizační techniku.

Podle získáných výsledků je možné vyzdvihnout určité přístupy, které se osvědčily při detekci anomálií na vybraných portech. Tyto doporučení pro detekci jsou shrnuty v následující tabulce 6.12. Hodnoty ve sloupci **A/C** označují buď ideální hodnotu `alfa` nebo ideální počet shluků. Ve sloupci **Kombinace** jsou vypsány vhodné kombinace vektorů vlastností a normalizací.

	Shlukování	**A/C**	**Kombinace**	**Poznámka**
TCP 22 (SSH)	Hierarch.	3	V1-P, V2-P	Velmi nízký a stabilní počet falešných detekcí pro všechny testované časy.
TCP 25 (SMTP)	Hierarch.	5+	V1-SP, V2-SP	Vzhledem k testovacím datům je toto jen hrubý odhad.
TCP 80 (HTTP)	Hierarch.	2	V1-M, V2-M	Pro větší počty hostů se více vyplatí rychlejší shlukování pomocí k-means.
UDP 53 (DNS)	K-means	10	V1-P, V2-P	
Protokol ICMP	K-means	5-	V1-P, V2-P	Nehrozí generování nadměrného počtu falešných detekcí.

Tabulka 6.12: Výběr nejlepších přístupů detekce anomálií pro použitá testovací data.

Kapitola 7

Závěr

V úvodu práce byl popsán způsob získávání dat o síťovém provozu pomocí NetFlow záznamů. Dále byly popsány základní metody detekce anomálií, z nichž se tato práce zaměřuje na metody shlukové analýzy. Byly vybrány a vysvětleny metody hierarchického a k-means shlukování. Kromě dvou metod shlukování se pracuje se dvěma vektory vlastností a s pěti technikami normalizace. Pro účely testování byly vybrány normalizace pomocí percentilů, průměru a směrodatné odchylky a odmocnění. Testování proběhlo nad daty síťového provozu z univerzitní sítě a výsledky byly prezentovány v přehledných tabulkách, které zobrazují počty detekovaných anomálií pro jednotlivé kombinace.

Testování ukázalo, že jednotlivé shlukovací metody a kombinace normalizací mají velmi podobné detekční schopnosti a nelze tedy jednoznačně určit ideální variantu. Kvalita detekce je ovlivněna charakterem vstupních dat a hlavně hodnotami parametrů shlukovacích algoritmů (alfa, počet shluků). Přesto bylo možné na základě výsledků testů určit přístupy, které by mohly být vhodnější než ostatní i pro jiné statistiky hostů. Ukázalo se, že pro nižší počty hostů hierarchické shlukování generuje méně falešných detekcí než k-means.

I přes časovou a paměťovou náročnost hierarchického shlukování bylo díky aplikaci jednoduchého filtrování dat možné dosáhnout přijatelných rychlostí výpočtu – maximálně 2 minuty. Po odečtení času potřebného pro zpracování 5-ti minutového NetFlow záznamu lze zbývající čas (než je třeba zpracovat dalších 5 minut provozu) využít pro spuštění dalších programů, které mohou nad detekovanými IP adresami spustit různé heuristické funkce, a tím zpřesnit detekci.

Literatura

[1] B. Claise, E.: Cisco Systems NetFlow Services Export Version 9. Dostupné z: `http://www.ietf.org/rfc/rfc3954.txt`, Říjen 2004.

[2] B. Claise, E.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. Dostupné z: `http://tools.ietf.org/html/rfc5101`, Leden 2008.

[3] Carter, K. M.; Lippmann, R. P.; Boyer, S. W.: Temporally oblivious anomaly detection on large networks using functional peers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, ACM, 2010, s. 465–471.

[4] Haag, P.: Nfdump [online]. Dostupné z: `http://nfdump.sourceforge.net`, 2010 [cit. 2013-03-22].

[5] Lakhina, A.; Crovella, M.; Diot, C.: Characterization of network-wide anomalies in traffic flows. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ACM, 2004, s. 201–206.

[6] Leung, K.; Leckie, C.: Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, Australian Computer Society, Inc., 2005, s. 333–342.

[7] Maimon, O. Z.; Rokach, L.: *Data Mining and Knowledge Discovery Handbook*. Springer, 2005.

[8] Münz, G.; Li, S.; Carle, G.: Traffic anomaly detection using k-means clustering. *Proc. of Leistungs-, Zuverlässigkeits-und Verlässlichkeitsbewertung von Kommunikationsnetzen und Verteilten Systemen*, ročník 4, 2007.

[9] Patcha, A.; Park, J.-M.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, ročník 51, č. 12, 2007: s. 3448–3470.

[10] StatSoft: Electronic Statistics Textbook [online]. Dostupné z: `http://www.statsoft.com/textbook/cluster-analysis`, 2013 [cit. 2013-04-17].

[11] Thottan, M.; Liu, G.; Ji, C.: Anomaly detection approaches for communication networks. In *Algorithms for Next Generation Networks*, Springer, 2010, s. 239–261.

[12] Wikipedia: NetFlow [online]. Dostupné z: `http://cs.wikipedia.org/wiki/Netflow`, 2013-03-14 [cit. 2013-03-22].

[13] Wikipedia: Standard score [online]. Dostupné z:
http://en.wikipedia.org/wiki/Standard_score, 2013-04-02 [cit. 2013-04-12].

Příloha A

Obsah CD

- Elektronická verze této práce ve formátu PDF.
- Zdrojové soubory této práce napsané pro LaTeX.
- Zdrojové kódy programu a soubory knihovny CLUTO.
- Soubor README s pokyny pro překlad a spuštění programu.
- Vybrané soubory se statistikami hostů.