VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IDENTIFIKACE POČÍTAČE POMOCÍ VZORŮ V SÍŤOVÉM PROVOZU

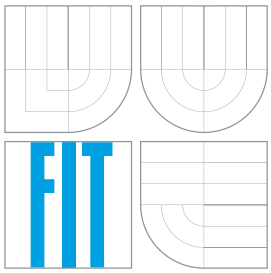
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAL MYŠKA

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IDENTIFIKACE POČÍTAČE POMOCÍ VZORŮ V SÍŤOVÉM PROVOZU

COMPUTER IDENTIFICATION BASED ON ITS NETWORK BEHAVIOUR

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAL MYŠKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LIBOR POLČÁK

BRNO 2014

Abstrakt

Tato diplomová práce se zabývá identifikaci počitačů s využitím vzorů chování v síťovém provozu. Jsou popsány bezpečnostní funkce zajišťující ochranu osobních údajů a bezpečnostní hrozby ohrožující soukromí uživatelů. Dále jsou popsány různé přístupy k identifikaci počítačů podle vzorů chování. Také je navržen nástroj pro identifikaci počítačů, využívající metodu pro dolování dat Multinomial Naive Bayes. Poté je popsána implementace navrženého nástroje a jsou provedeny experimenty zjišťující jeho úspěšnost identifikace.

Abstract

This diploma thesis deals with computer identification using network behavioral patterns. Security functions providing privacy are described together with user privacy threats. Then, several approaches to the computer identification based on network behaviour are described. The proposed tool is based on data mining method Multinomial Naive Bayes. Then, the implementation of proposed tool is described and the experiments recognizing success in the identification are performed.

Klíčová slova

Vzory chování, identifikace počítače, dolování dat, bezpečnostní hrozby

Keywords

Behavioral patterns, computer identification, data mining, security threats

Citace

Michal Myška: Identifikace počítače pomocí vzorů v síťovém provozu, diplomová práce, Brno, FIT VUT v Brně, 2014

Identifikace počítače pomocí vzorů v síťovém provozu

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Libora Polčáka.

.....
Michal Myška
27. května 2014

Poděkování

Chtěl bych poděkovat svému vedoucímu diplomové práce Ing. Liboru Polčákovi za odbornou pomoc, cenné připomínky a trpělivost při řešení této diplomové práce. Také bych chtěl poděkovat panu Ing. Matěji Grégrovi za poskytnutí záznamů Netflow.

Obsah

1 Úvod **3**

2 Behaviorální identifikace **5**
2.1 Soukromí a bezpečnostní funkce . . . 5
2.1.1 Anonymita . . . 5
2.1.2 Pseudonymita . . . 5
2.1.3 Nesledovatelnost . . . 6
2.1.4 Nedohledatelnost . . . 7
2.2 Identita . . . 8
2.2.1 Částečná identita . . . 8
2.2.2 Digitální identita . . . 8
2.3 Soukromí a hrozby . . . 9
2.4 Dolování dat . . . 9

3 Zpracování vstupních dat **11**
3.1 Vektorový model . . . 11
3.2 Atributy pro identifikaci . . . 11
3.3 Transformace atributů . . . 13

4 Algoritmy pro identifikaci počítače **15**
4.1 Multinomial Naive Bayes . . . 15
4.2 Měření podobnosti na základě Jaccardova koeficientu . . . 15
4.3 Kosinova podobnost . . . 16
4.4 Support Vector Machines . . . 16
4.5 Dosavadní využití dolovacích algoritmu pro identifikaci počítačů . . . 18
4.5.1 Multinomial Naive Bayes . . . 18
4.5.2 Kosinova podobnost . . . 19
4.5.3 Support Vector Machines . . . 19

5 Návrh nástroje pro identifikaci počítačů **20**
5.1 Výběr atributů . . . 20
5.2 Netflow . . . 21
5.2.1 Architektura Netflow . . . 21
5.2.2 Záznamy Netflow . . . 21
5.3 Vytváření modelu chování . . . 22
5.4 Průběh identifikace počítače . . . 24
5.5 Krajní případy při identifikaci počítačů . . . 25
5.5.1 Přiřazení referenčního počítače k více testovaným počítačům . . . 25

5.5.2 Testovaný počítač nelze nalézt v referenčním modelu 26
5.5.3 Odstranění cílové IP adresy z identifikace 26

6 Implementace **27**
6.1 Popis programu 27
6.2 Zpracování vstupních dat 30
6.3 Operace s vektorovými modely 31

7 Experimenty **33**
7.1 Experimenty s daty ze simulovaného prosředí 33
7.2 Experimenty s daty z reálného prosředí 35
7.2.1 Měření úspěšnosti pro různé porty 35
7.2.2 Určení prahu pro stanovení neúspěšného nalezení počítače 36
7.2.3 Doba trvání běhu programu 37
7.3 Zhodnocení výsledků 38

8 Závěr **39**

A Obsah CD **43**

Kapitola 1

Úvod

Internet v dnešní době představuje rozsáhlou síť využívanou velkým počtem uživatelů a podporující množství různorodých služeb. Které služby jsou uživateli využívané a jakým způsobem je využívají, blízko koresponduje se zájmy uživatelů. U některých služeb jako jsou například vyhledávače, uživatelé chtějí, aby služba nabízela pouze požadované informace, aniž by se jim zobrazoval nedůležitý obsah. Tyto služby často pracují s uživatelskými profily. Uživatelské profily obsahují informace o uživatelských aktivitách v minulosti. U webových služeb se můžeme setkat s *cookies*. *Cookies* obsahují informace o navštívených webových stránkách. Sledováním *cookies* dokáže pozorovatel sjednotit více uživatelských sezení dohromady. To umožňuje pozorovateli vytvářet uživatelské profily z webových stránek, se kterými uživatel komunikoval.

Informace o uživatelských aktivitách mohou být získány i na nižších vrstvách, zachycením síťového provozu na aktivních síťových prvcích. Informace o uživatelích se získávají pasivně. Při pasivním sběru informací nedochází k interakci s uživatelem. To představuje velkou hrozbu pro soukromí uživatelů, protože uživatelé nedokážou pasivní sběr informací detekovat. Z různých vlastností odvozených ze síťového provozu lze vytvořit uživatelské profily. Tyto profily mohou být použity k dohledání uživatelovy identity.

Jednou z možností dohledání identity uživatele je identifikace podle vzorů chování. Tato technika je založena na hledání podobností v minulých aktivitách uživatele. Základem identifikace podle vzorů chování je sloučení vlastností síťového provozu daného uživatele a následné zpracování získaných dat. K těmto činnostem lze dobře využit techniky pro dolování dat. V širším pojetí technika dolování dat představuje hledání vzorů v získaných datech, analýzu těchto vzorů z různých pohledů, jejich kategorizaci a sumarizaci, pro získání užitečných informací.

Tato diplomová práce se zabývá různými metodami identifikace podle vzorů chování a jejich dopadem na soukromí uživatelů. Cílem této práce je návrh a implementace nástroje, který dokáže identifikovat uživatele pomocí extrahovaných atributů ze síťového provozu. Součásti této práce je také provedení experimentů s vyvinutým nástrojem a zhodnocení jeho vlatností jako jsou úspěšnost nebo rychlost zpracování.

V následující kapitole této práce jsou popsány základní bezpečnostní funkce anonymita, pseudonymita, nesledovatelnost a nedohledatelnost. Dále je definován pojem identita a vypsány bezpečnostní hrozby. Také je zde detailněji popsán pojem dolování dat, a jak může ohrožovat soukromí uživatelů. Ve třetí kapitole je popsán model pro reprezentaci zpracovaných dat. Jsou zde popsány atributy využitelné pro indentifikaci a transformace atributů, které je možné použít pro zpřesnění výsledné identifikace. Současně je ve třetí kapitole popsán protokol Netflow, komponenty jeho architektury a také jsou v této kapitole uvedeny

ukládané atributy síťového provozu. Čtvrtá kapitola popisuje vybrané dolovací metody, použitelné pro identifikaci počítačů. V páté kapitole je popsán návrh nástroje pro identifikaci. V šesté kapitole je popsána implementace navrženého nástroje. V sedmé kapitole jsou popsány provedené experimenty. V závěru jsou shrnuty a zhodnoceny výsledky této práce. Dále jsou zde popsány možnosti navázání na tuto práci. Tato diplomová práce vychází ze semestrálního projektu, ve kterém jsem se seznámil s problematikou identifikace počítačů a navrhl nástroj pro identifikaci počítačů. Ze semestrálního projektu byly převzaty kapitoly 2, 3, 4 a část kapitoly 5.

Kapitola 2

Behaviorální identifikace

Jelikož identifikace počítačů patří mezi hrozby pro soukromí uživatelů je tato kapitola věnována teorii z oblasti bezpečnosti a ochrany soukromí. V první podkapitole jsou popsány základní bezpečnostní funkce anonymita, pseudonymita, nesledovatelnost a nedohledatelnost. Dále jsou vypsány bezpečnostní hrozby a jejich rozdělení do tříd. Také je vysvětlen pojem identita a jaké může mít podoby. V poslední části je popsána technika dolování dat a její dopad na soukromí uživatelů.

2.1 Soukromí a bezpečnostní funkce

V této podkapitole jsou popsány definice [19] anonymity, pseudonymity, nesledovatelnosti, nedohledatelnosti. Tyto základní bezpečnostní funkce mají blízký vztah a mohou být používány na základě hrozeb pro soukromí uživatelů. Také jsou popsány některé bezpečnostní mechanizmy, které tyto bezpečnostní funkce zajišťují.

2.1.1 Anonymita

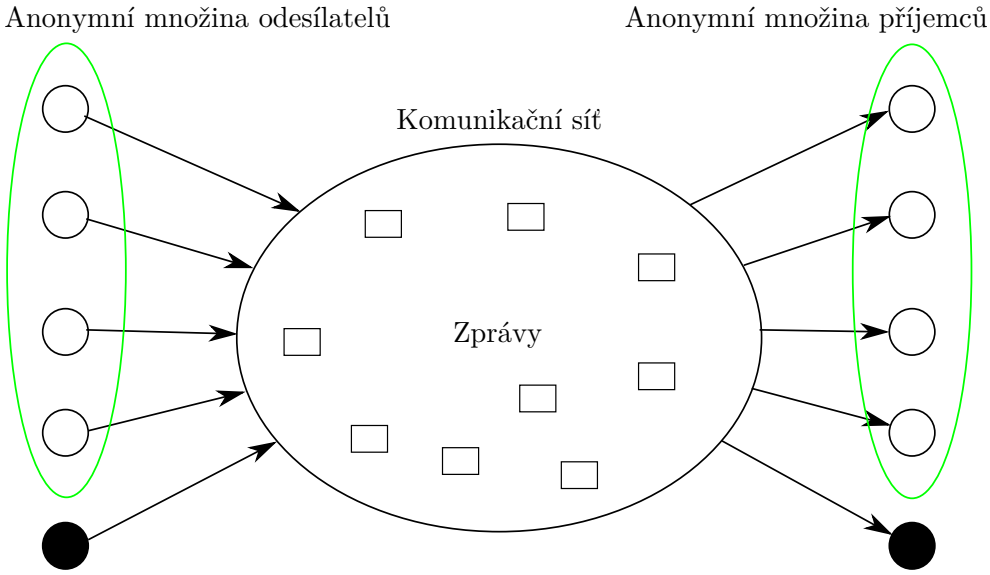
K zajištění anonymity je potřeba nalézt více uživatelů s potenciálně stejnými vlastnostmi. Anonymita je zajistitelná v rámci množiny těchto uživatelů, označovanou jako anonymní množina.

Anonymní množina [19] je množina všech uživatelů se stejnými atributy. V rámci dané anonymní množiny se mohou vyskytovat pouze uživatelé, kteří mohou provést stejnou operaci. Příklad anonymních množin pro komunikaci odesílatelů a příjemců je na obrázku 2.1.

Anonymita je definována jako stav, kdy uživatel není identifikovatelný v rámci anonymní množiny. Anonymita zajišťuje možnost využití zdrojů nebo služeb bez toho, aby byla odhalena uživatelova identita. To zahrnuje také nemožnost sledovat stopy vedoucí k uživateli. Pokud útočník bude chtít na základě získané informace dohledat anonymního uživatele, informace povede ke všem uživatelům v rámci anonymní množiny [2].

2.1.2 Pseudonymita

Pseudonymita [19] uživatele představuje použití pseudonymu místo skutečného jména. Pseudonym je identifikátor, pod kterým uživatel vystupuje a má podobu posloupnosti znaků. Pseudonym nesmí obsahovat informace o vazbě s pravou identitou uživatele nebo informace o pravé identitě uživatele. Na obrázku 2.2 je ukázán příklad využití pseudonymu v rámci komunikace odesílatelů a příjemců.



Obrázek 2.1: Anonymita zajištěna pomocí anonymních množin [19]. Bíle kolečko je anonymizovaný uživatel, černé kolečko je neanonymizovaný uživatel a čtverec je zasílána zpráva.

Pseudonymita stejně jako anonymita zajišťuje ochranu pravé identity uživatele. Pseudonymita navíc umožnuje využívat služby nebo zdroje, jejichž využití je účtované.

U pseudonymity není specifikováno, do jaké míry je známý vztah mezi pseudonymem a identitou uživatele. Z pohledu úrovně znalosti vztahu mezi pseudonymem a pravou identitou můžeme pseudonymy rozdělit na několik typů:

Veřejný pseudonym

U veřejného pseudonymu je znám vztah mezi pseudonymem a identitou uživatele. Veřejný pseudonym může být například telefonní číslo ve spojení s jeho vlastníkem [19].

Neveřejný pseudonym

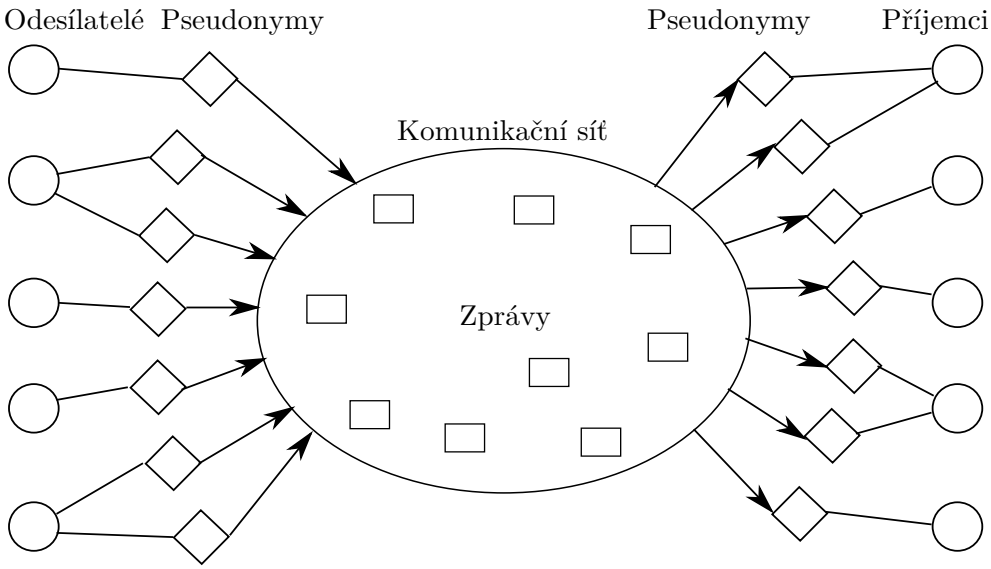
U neveřejného pseudonymu mohou znát vztah mezi uživatelem a pseudonymem jen některé subjekty. Tento vztah ale není veřejně známý. Příklad neveřejného pseudonymu může být číslo bankovního účtu, kdy banky znají vlastníka tohoto účtu [19].

Nedohledatelný pseudonym

U Nedohledatelného pseudonymu není nikomu znám vztah mezi pseudonymem a pravou identitou, kromě samotného vlastníka pseudonymu [19].

2.1.3 Nesledovatelnost

Nesledovatelnost [2] uživatele zajišťuje, že při využití služeb nebo zdrojů nejsou ostatní uživatelé schopní dohledat jeho aktivitu. Stejně jako jsme měli anonymní množinu v podkapitole 2.1.1, v rámci které byla zajištěna anonymita, také máme i nesledovatelnou množinu zajišťující nesledovatelnost. Na obrázku 2.3 je příklad nesledovatelných množin pro komunikaci odesílatelů a příjemců.



Obrázek 2.2: Pseudonymita zajištěna pomocí pseudonymů [19].

Nesledovatelnost uživatele je dosažena ukrytím zdrojů a služeb a ne ukrytím identity uživatele. Toho lze docílit několika bezpečnostními mechanismy:

- Alokace informací zajišťující nesledovatelnost:

  Služby nebo zdroje jsou poskytovány na více místech. Když uživatel chce danou službu využít, náhodně si vybere jednu lokaci, kde jsou poskytovány. Tento bezpečnostní mechanismus vyžaduje, aby služba nebo zdroj nemohla být pozorována jinými uživateli [2].

- Všesměrové vysílání:

  Dalším možným mechanismem je všesměrové vysílání. Pokud je zpráva z dané služby nebo zdroje poslána všem, není možné zjistit, kdo chce přijmout a využít danou zprávu [2].

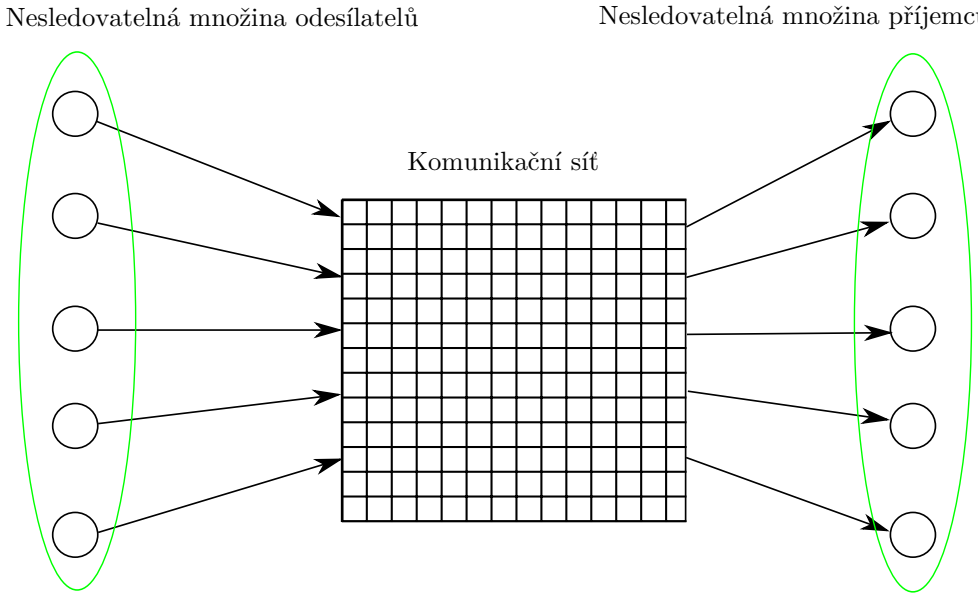
- Šifrování:

  Pokud útočník sleduje tok zpráv, může získat informace z atributů zprávy nebo z faktu, že zpráva byla poslána. Tomu lze zabránit použitím šifrování zprávy nebo posíláním náhodných zpráv v průběhu komunikace, které s přenášenými informacemi nijak nesouvisí [2].

2.1.4 Nedohledatelnost

Nedohledatelnost [2] uživatele zajišťuje, že pokud uživatel opakovaně využívá různé služby, tak ostatní uživatelé nejsou schopni dohledat spojitosti mezi akcemi, které tento uživatel provedl. A tedy nedohledatelnosti opakovaného užití stejné služby i nedohledatelnost užití různých služeb.

Základní požadavek pro dodržení nedohledatelnosti je ochrana identity uživatele před analýzou jeho chování. Toho lze docílit anonymitou. Další možnosti je použití více pseudonymů, u kterých není znám vztah mezi pseudonymem a uživatelovou pravou identitou.



Obrázek 2.3: Nesledovatelnost zajištěna pomocí nesledovatelných množin [19].

2.2 Identita

Identita [19] je libovolná podmnožina všech atributů jednoho uživatele, která přesně vystihuje právě tohoto uživatele. Atributy představují charakteristické rysy, kterými jsme schopni daného uživatele popsat. Identita nemusí být vztažena jen k lidské osobě, ale i k libovolnému subjektu jako je například počítač. V rámci této práce budeme mluvit o identitě vztahující se k uživateli.

Uživatel nemusí mít pouze jednu identitu, ale několik. V tomto kontextu můžeme mluvit i o úplné identitě, která představuje sjednocení všech hodnot atributů, představující daného uživatele.

Z pohledu útočníka lze také definovat identifikovatelnost uživatele: identifikovatelnost uživatele znamená, že útočník může jednoznačně určit daného uživatele ve skupině různých uživatelů.

Identifikace uživatele nemusí být vždy úspěšná, protože některé hodnoty atributů se mohou překrývat s hodnotami atributů jiného uživatele. Hodnoty atributů uživatele se také mohou v průběhu času měnit, proto některá identita uživatele nemusí být po určité době stejná.

2.2.1 Částečná identita

Identita uživatele může zahrnovat více částečných identit. Částečná identita je podmnožina všech atributů, které představují uživatele ve specifickém kontextu nebo roli.

Jako identifikátor částečné identity může být použit pseudonym. Ten umožňuje navázání na provedené operace v daném kontextu nebo roli, jako je například autentizace [19].

2.2.2 Digitální identita

Digitální identita uživatele představuje přiřazení hodnot atributů k danému uživateli, které mohou být dostupné v rámci počítačových systému. Může to být například e-mail nebo

přihlašovací jméno. Digitální identita by měla slučovat všechna data uživatele, která mohou být uložena a vzájemně propojena počítačovými systémy [19].

2.3 Soukromí a hrozby

Tato podkapitola se zabývá hrozbami, které mohou ohrozit soukromí uživatelů. Solove [21] se zabývá soukromím a jeho hrozbami z právního pohledu. Zasažené subjekty mohou být například osoby nebo organizace. Solove definuje hrozby pro soukromí subjektu a rozděluje je do čtyř tříd:

- Sběr informací:

  Do této třídy patří sledování a výslech. I když nedochází ke zveřejnění, shromažďování informací může ohrozit soukromí subjektu.

- Zpracování informací:

  Tato třída zahrnuje aktivity, které již pracují se získanými informacemi. Tyto aktivity zahrnují způsoby, jakým jsou informace udržovány a používány. Patří zde agregace, identifikace, nejistota, znoupoužití a vyloučení.

- Šíření informací:

  Všechny hrozby v této třídě zahrnují zveřejnění nebo předání soukromých informací třetí straně. Do této třídy patří porušení důvěrnosti, zveřejnění, odhalení, zvýšená dostupnost, vydírání, přivlastnění a zkreslení.

- Invaze:

  Tato třída se liší od předchozích, protože nezahrnuje soukromé informace o subjektu. Hrozby v této třídě se vyznačují aktivním zasahováním do soukromí. Nejde tedy o získávání informací od subjektů, ale invazivní procesy směřující na subjekty. Do této třídy patří obtěžování a omezení rozhodování.

Na obrázku 2.4 je zobrazen vztah mezi výše popsanými třídami. Datový sklad představuje entitu provádějící sběr, zpracování a šíření informací.

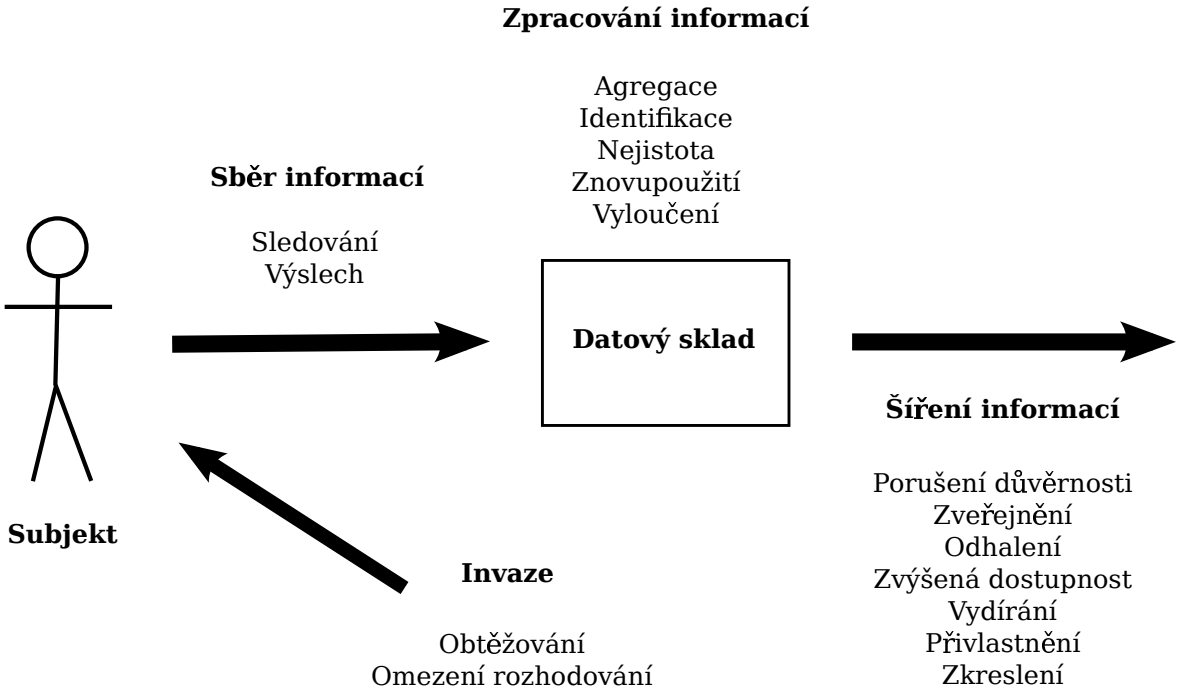
2.4 Dolování dat

Dolování dat se stalo velmi důležitou disciplínou, když se začaly ve velkém množství sbírat a ukládat nejrůznější data a začaly vznikat velké databáze [10].

Dolování dat [22] je technika hledání vzorů a různých vztahů ve velkých objemech dat. Kombinuje techniky umělé inteligence, statistiky, rozpoznávání vzorů, datové vizualizace a znalosti získané z expertních systémů.

Cavoukian [7] definuje dolování dat jako množinu automizovaných technik použitých k extrakci skrytých nebo původně neznámých informací z velkých databází.

Technika dolování dat se používá v různých oblastech. My se zaměřujeme na oblast dat získáných z oblasti webu. V dolování informací ve webových datech existují tři hlavní směry [5]:



Obrázek 2.4: Vztah mezi jednotlivými třídami hrozeb [21].

- Dolování informací:

  Dolování informací se zaměřuje na vývoj technik, pomáhající uživatelům zpracovat velké množství dat během vyhledávání a najít informace, které uživatel potřebuje.

- Dolování struktury webových odkazů:

  Dolování struktury webových odkazů se zaměřuje na vývoj technik pro zlepšení vyhodnocování kvality webových stránek ve formě hypertextových odkazů.

- Dolování vzorů chování:

  Tento směr se zabývá vývojem technik pro studování a vyhodnocování chování uživatele při procházení webových stránek. Znalosti vzorů chování uživatelu může například umožňovat přizpůsobit rozhraní webových stránek pro každého uživatele.

Tavani se také zabýval, jak dolování dat ovlivňuje soukromí uživatelů [22]. Ohrožení soukromí přichází již s novými technologiemi, které často umožňují sběr informací o jednotlivých uživatelích nebo o skupinách, bez toho, aby byli o tom daní uživatelé informováni. V některých případech jsou uživatelé informováni o sběru informací skrze danou technologii, ale již nejsou informování o tom, jak budou tyto informace využity. To platí i v případě dolování dat. Data jsou často sbírána bez vědomí uživatelů. V případě, že jsou uživatelé informováni, tak není možné předpovědět, jaké nové informace dolovací algoritmy objeví.

Kapitola 3

Zpracování vstupních dat

V první podkapitole této sekce je popsán vektorový model vhodný pro ukládání dat obsahující aktivitu uživatelů. Vektorový model již byl úspěšně použit při identifikaci uživatelů jako struktura pro ukládání vstupních dat [11]. Dále jsou popsány různé atributy použitelné pro identifikaci podle vzorů chování. V poslední části jsou popsány transformace atributů, které je možné použít k zpřesnění výsledné identifikace.

3.1 Vektorový model

Vektorový model je algebraická struktura používaná pro reprezentaci textu [17]. Pro jednotlivé výrazy se mohou uchovávat různé atributy, jako je například pozice v textu. Jeden dokument je reprezentován jedním vektorem. Vektor je vytvořen jako skalární součin atributů pro každý výraz v dokumentu. Typickým atributem je například počet výrazů v jednom dokumentu. Všechny dokumenty tedy jsou reprezenovány jako množina vektorů [17].

Na obrázku 3.1 [20] je ukázán příklad trojrozměrného vektorového modelu v prostoru pro reprezentaci textového dokumentu. Proměnná D_i značí vektor dokumentu skládajícího se z hodnot atributů pro výrazy T_i.

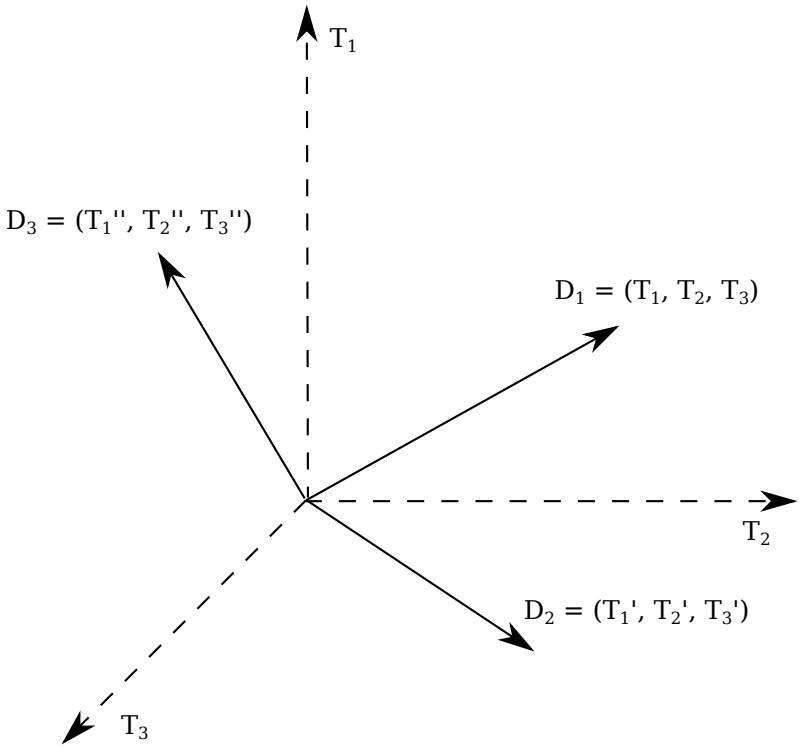
Tuto strukturu lze využít i pro modelování chování uživatelů. Jeden uživatel je reprezentován více vektory s atributy pro jednotlivé cílové počítače, se kterými uživatel komunikuje. Ukládané atributy mohou být například počet přenesených paketů mezi uživatelem a cílovým počítačem nebo doba navázané komunikace.

3.2 Atributy pro identifikaci

Jedním z důležitých kroků při dolování dat je výběr atributů pro identifikaci. Některé atributy lze přímo získat z datových toku, některé musí být odvozeny. Pro identifikaci je možné použít jeden vybraný atribut nebo jejich kombinaci. Vhodný výběr atributů může usnadnit klasifikaci a zpřesnit výsledné hodnoty. Panchenko a Niessen [18] navrhli a využili pro identifikaci tyto atributy:

- Počet přenesených paketů:

  Pro každý datový tok je sečten počet paketů. Tento atribut je možné nadále upravovat například normalizaci nebo zaokrouhlováním.



Obrázek 3.1: Ilustrace vektorového modelu v prostoru [20].

- Procento příchozích paketů:

  Pro tento atribut se měří počet paketů v každém směru. Poměr příchozích paketů může sloužit také jako atribut pro identifikaci počítačů.

- Velikost paketů:

  Pro každý datový tok se počítá počet paketů vyskytujících se velikostí. Velikost paketů se může počítat i pro každý směr toku zvlášť.

- Počet přenesených bytů:

  Tento atribut je získán součtem velikosti všech paketů. Je možné počítat počet přenesených bytů také pro každý směr toku zvlášť.

- Značky velikosti datového toku:

  Značka představuje označení místa v datovém toku, kde se mění směr přenesených paketů. Při každé změne směru se uloží hodnota, kolik paketů bylo doposud přeneseno v daném směru. Poté jsou sečteny velikosti všech paketů v daném směru a tím získáme značky velikosti datového toku.

Je však možné použít i další atributy. Dyer a Coull [9] využili pro identifikaci tyto atributy:

- Celkový čas komunikace:

  Další atribut použitelný pro identifikaci je celkový čas komunikace. Jedná se o méně užitečný atribut [9], který sám o sobě neposkytuje příliš velkou diverzi jednotlivých datových toků.

- Šířka pásma:

  Pro každý směr můžeme určit spotřebovanou šířku pásma pro komunikaci. Tento atribut si zachovává poměrně velkou diverzitu i při aplikaci opatření proti identifikaci.

3.3 Transformace atributů

Herrman et al. [11, 4, 12] využil transformace typu Inverse document frequency, Term frequency a Kosinova normalizace pro zpřesnění identifikace počítačů. Transformace aplikoval na hodnoty atributů uložených ve vektorovém modelu, popsaném v podkapitole 3.1. Tyto transformace se také používají pro zpřesnění klasifikace u dolování dat z textu [25].

- Transformace typu inverse document frequency:

  Všechny atributy nacházející se ve vektorech jsou klasifikačním algoritmem pokládany za stejně významné. Pokud se atribut nachází v každém vektoru, neposkytuje tolik informací jako atribut s menší četnosti výskytu. Inverse document frequency [17, 15] posoudí důležitost daného atributu na základě jeho četnosti výskytu ve vektorech. Na základě důležitosti daného atributu se přidá ohodnocení. Toto ohodnocení označíme I_t a jeho výpočet je dán vzorcem 3.1.

$$I_t = log(N/f_t) \tag{3.1}$$

  Proměnná f_t představuje počet vektorů, které obsahují daný atribut t. Proměnná N ve vzorci představuje celkový počet vektorů.

- Transformace typu term frequency:

  Atributy s velkými hodnotami mohou zastínit relevanci ostatních atributů vektorů a tím snížit rozlišitelnost mezi jednotlivými vektory. Například cílový počítač s deseti přístupy není desetkrát důležitější než počítač s jedním přístupem. Tento problém lze vyřešit sublineární transformaci všech atributů pomocí vzorce 3.2:

$$f'_{t,d} = 1 + log(f_{t,d}) \tag{3.2}$$

  Tím získáme nový počet výskytů označený $f'_{t,d}$. Proměnná $f_{t,d}$ představuje původní počet výskytu atributu t ve vektoru d [11].

- Transformace typu term frequency - inverse document frequency:

  Je možné předchozí transformace typu term frequency a inverse document frequency zkombinovat a díky tomu vytvořit ohodnocení pro každý atribut ve vektoru. Toto ohodnocení označené jako $w_{t,d}$ lze získat pomocí vzorce 3.3:

$$w_{t,d} = (1 + log(f_{t,d})) \cdot log(N/f_t) \tag{3.3}$$

  Proměnná $f_{t,d}$ značí četnost výskytů atributu t ve vektoru d. f_t představuje počet výskytů daného atributu v jednotlivých vektorech t a N je celkový počet vektorů [15].

- Kosinova normalizace:

  Další možnou heuristikou je kosinova normalizace [11, 15], použita pro zvýšení přesnosti klasifikačních algoritmů nebo algoritmů pro dolování informací [17]. U vektorových modelů může být použita na jednotlivé atributy ve vektorech, tím, že každý atribut vydělí Euklidovou vzdáleností daného vektoru dle vzorce 3.4:

$$f_{v,i}^{norm} = \frac{f_{v,i}}{||d_v||} \tag{3.4}$$

  Proměnná $f_{v,i}$ značí atribut vektoru v a proměnná $||d_v||$ je Euklidova vzdálenost vektoru v, která lze získat pomocí vzorce 3.5:

$$||d_v|| = \sqrt{\sum_{i=1}^{n} f_{v,i}^2} \tag{3.5}$$

Kapitola 4

Algoritmy pro identifikaci počítače

V předchozí kapitole byly uvedeny atributy použitelné pro identifikaci počítačů a struktura pro jejich uchovávání. V této kapitole jsou uvedeny vybrané metody pro identifikaci počítačů. Jedná se o metody určené k dolování dat, které byly úspěšně použity k identifikaci počítačů podle vzorů chování. V poslední podkapitole je popsáno konkrétní použití dolovacích metod pro identifikaci počítačů.

4.1 Multinomial Naive Bayes

Klasifikační metoda Multinomial Naive Bayes [25] je původně určena pro kategorizaci textu, založena na pravděpodobnostní příslušnosti do dané třídy. Jelikož identifikace počítačů může být považována za klasifikační problém, byla již tato metoda zdárně použita k tomuto účelu [12, 4, 11].

Mějme instanci d náležící do množiny všech možných instancí X. Také mějme pevně daný počet klasifikačních tříd c. Určení pravděpodobnosti, že instance d patří do dané třídy c je dána vzorcem 4.1:

$$P(c|d) \approx \prod_{h \in H} P(h|c)^{f_{h,d}} \tag{4.1}$$

$P(h|c)$ je pravděpodobnost, že atribut h náležící do instance d je nalezen ve třídě c. Proměnná $f_{h,d}$ je frekvence výskytu atributu h v instanci d. Poté se přiřadí dané třídě instance d s největší pravděpodobností $P(c|d)$ [4].

4.2 Měření podobnosti na základě Jaccardova koeficientu

Jaccardův koeficient je metrika pro nalezení podobnosti dvou množin. Liberatore a Levine [16] použili jaccardův koeficient pro identifikaci webových stránek. Pro webové stránky měli minimálně jeden záznam odchycené komunikace. Tyto záznamy měly přidělenou adresu webové stránky a byly použity pro trénování. Dále měli záznamy komunikace, u kterých webové stránky nebyly známy. Ty byly použity pro testování. Pro každou komunikaci s webovou stránkou vytvořili instanci obsahující atributy velikost a směr každého paketu.

Každá webová stránka byla reprezentována jako množina dvojic atributů velikost a směr paketu. Pokud pro webovou stránku existovala pouze jedna trénovací instance, do množiny představující danou webovou stránku byly vloženy všechny dvojice atributů z trénovací

instance. Pokud pro webovou stránku existovalo více trénovacích instancí, byly do množiny vloženy jen dvojice atributů, které se vyskytovaly ve většině trénovacích instancí.

Jaccardův koeficient dvou množin A, B poté získáli pomocí vzorce:

$$S_{A,B} = \frac{|A \cap B|}{|A \cup B|} \tag{4.2}$$

Proměnna A představuje množinu dvojic reprezentující známé webové stránky a proměnna B je množina dvojic z testovací instance bez přidělené webové adresy. Přiřazení testovací instance k webové stránce se provede nalezením největšího jaccardova koeficientu.

4.3 Kosinova podobnost

Kosinova podobnost je založena na porovnání dvou vektorů, kdy jejich podobnost je dána kosinem úhlu, který dané vektory svírají. Pro dva vektory $A = (a_1, a_2, ..., a_n)$ a $B = (b_1, b_2, ..., b_n)$ lze Kosinovu podobnost zapsat vzorcem 4.3:

$$cos(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{4.3}$$

Čitatel $A \cdot B$ je skalární součin vektorů dle vzorce 4.4:

$$A \cdot B = \sum_{i=1}^{n} a_i \cdot b_i \tag{4.4}$$

Jmenovatel $||A|| \cdot ||B||$ je součin normovaných vektorů A, B dle vzorce 4.5:

$$||A|| \cdot ||B|| = \sqrt{\sum_{i=1}^{n} a_i^2} \cdot \sqrt{\sum_{i=1}^{n} b_i^2} \tag{4.5}$$

A tedy konečný vzorec Kosinovy podobnosti je:

$$cos(A, B) = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \cdot \sqrt{\sum_{i=1}^{n} b_i^2}} \tag{4.6}$$

Výstupní hodnoty Kosinovy podobnosti jsou v rozsahu nula až jedna, kdy nula značí, že vektory si nejsou vůbec podobné a jedna značí, že vektory jsou totožné [15].

4.4 Support Vector Machines

Support Vector Machines je rodina učebních metod, pro analýzu dat a rozpoznávání vzorů, používané často při dolování dat. Tyto metody se vyznačují velkou přesností klasifikace.

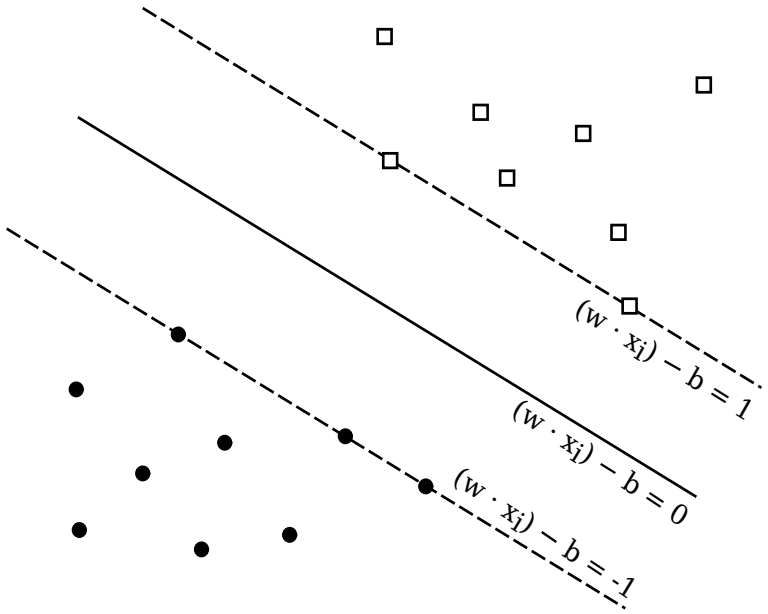
Základní myšlenka je reprezentace tříd pomocí vektorů ve vektorovém prostoru. V rámci identifikace podle vzorů chování tyto vektory představují atributy datových toků. Na základě trénovacích dat se klasifikátor snaží nalézt co největší lineárně separovaný prostor mezi dvěma třídami. Support vectors jsou hodnoty atributů z trénovací množiny, představující hranice mezi jednotlivými třídami. Jsou to nejdůležitější hodnoty pro zařazování objektů do jednotlivých tříd.

V případě, že vektory nejsou lineárně separovatelné, je potřeba nalézt mapování z nelineárního prostoru do lineárního. To lze zajistit převodem vektorového prostoru na vícedimenzionální prostor [6, 18].

U Support Vector Machines se vytváří bínární model, který dokáže klasifikovat testovací data do dvou tříd. Z trénovacích dat vytváříme dvojice $(x_i, y_i), i = 1, ..., l$, u kterých $x_i \in R^n$ jsou vstupní vektory atributů a $y_i \in \{-1, 1\}$ je označení třídy, do které vektory patří. Ve vícedimenzionalním prostoru jsou dvě třídy odděleny pomocí nadroviny definované pomocí vzorců 4.7.

$$\begin{aligned} (w \cdot x_i) - b \geq 1 \text{ pokud } y_i = 1 \\ (w \cdot x_i) - b \leq -1 \text{ pokud } y_i = -1 \end{aligned} \tag{4.7}$$

Proměnná w je normála k nadrovině [23]. Příklad nadroviny oddělující dvě třídy je znázorněn na obrázku 4.1.



Obrázek 4.1: Grafické znázornění nadroviny oddělující dvě třídy [17].

V rámci Support Vector Machines se řeší binární klasifikační problém.

$$min(\frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i)$$

$$\text{podléhá } y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \text{ pro } \xi_i \geq 0 \tag{4.8}$$

Proměnná $C > 0$ je vstupní parametr klasifikátoru, který slouží k snížení počtu chyb při učení klasifikátoru. ξ_i je proměnná určující odchylku chybné klasifikace hodnoty x_i. Funkce ϕ slouží k namapování vstupních vektorů x_i do vícedimenzionalního prostoru [13]. Klasifikační problém popsaný ve vzorci 4.8 lze reprezentovat duální formou [24].

$$W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \tag{4.9}$$

Pro rovnici zapsanou duální formu 4.9 platí omezení.

$$0 \leq \alpha_i \leq C, i = 1, ..., l$$
$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{4.10}$$

Proměnná α_i je Lagrangeův multiplikátor [17]. $K(x_i, x_j)$ je jádrová funkce. Jádrové funkce mapují vstupní data do vícedimenzionalního prostoru, aby bylo možné atributy dvou tříd lineárně oddělit. Mezi základní typy jádrových funkí patří:

- Lineární funkce: $K(x_i, x_j) = x_i^T \cdot x_j$
- Polynomiální funkce: $K(x_i, x_j) = (\gamma \cdot x_i^T \cdot x_j + r)^d$ pro $\gamma > 0$
- Radiální bázová funkce: $K(x_i, x_j) = exp(-\gamma \cdot ||x_i - x_j||^2)$ pro $\gamma > 0$

Proměnné r, d, γ jsou vstupní parametry jádrových funkcí. Support Vector Machines jsou navrženy pouze pro binární klasifikaci, to znamená klasifikaci pouze mezi dvěma třídami. Pro klasifikaci mezi více třídami je potřeba vytvořit nové metody. Jedno z možných řešení je sestavení sérií binárních klasifikátorů.

- Jeden proti všem [14]:

  V rámci metody *jeden proti všem* se pro k tříd vytváří k binárních klasifikátorů. Pro každou třídu je sestaven klasifikátor, u kterého se vytvoří nadrovina oddělující prostor dané třídy od prostoru všech ostatních tříd. Při klasifikaci pro k tříd se porovnávají výsledky všech binárních klasifikátorů a vybere se třída, pro kterou rozhodovací funkce měla největší hodnotu.

- Jeden proti jednomu [14]:

  U metody *jeden proti jednomu* se pro k tříd vytváří $k \cdot (k-1)/2$ binárních klasifikátorů. Každý klasifikátor je výtvářen pro každou dvojici tříd. V rámci této metody můžou být použity různé typy rozhodovacích strategií pro výběr správné třídy. Hsu a Lin [14] využili rozhodovací strategii založenou na výběru nejčastěji volené třídy. V rámci binárního klasifikátoru obdrží vybraná třída jeden hlas. Nakonec je z k tříd vybrána třídá, která po provedení $k \cdot (k-1)/2$ binárních klasifikaci měla nejvíce hlasů.

4.5 Dosavadní využití dolovacích algoritmu pro identifikaci počítačů

V této podkapitole jsou popsány případy využití vybraných dolovacích metod pro identifikaci počítačů. U každého případu jsou popsány vstupní data a způsob, jakým byly zpracovány. Nakonec jsou popsány výsledky, které byly u jednotlivých metod dosaženy.

4.5.1 Multinomial Naive Bayes

Banse, Herrmann a Federrath [4] využili pro identifikaci počítačů algoritmus Multinomial Naive Bayes, vysvětlený v podkapitole 4.1. Jako vstupní data měli k dispozici DNS dotazy získané pomocí DNS resolveru z univerzitní sítě. Vstupní data měli k dispozici z časového

úseku od 2. února 2010 do 30. července 2010. Ze získaných dat měli dohromady 18 904 počítačů s unikátní zdrojovou IP adresou.

Jako atributy pro identifikaci využili cílovou IP adresu a počet navázaných spojení. Data rozdělovali do trénovacích instancí po časových úsecích 24 hodin. Pro zpřesnění identifikace využili transformaci typu term frequency - inverse document frequency, popsanou v podkapitole 3.3.

Při identifikaci sledovali úspěšnost pro různé množství použitých trénovacích instancí. Úspěšnost identifikace se pohybovala v rozmezí 69.2% až 89%. Také sledovali závislost úspěšnosti na stáří trénovací instance od pěti do devadesáti dnů. Pro pět dnů starou instanci byla úspěšnost identifikace 78.3% a pro devadesát dnů starou instanci byla úspěšnost identifikace 44.7%.

4.5.2 Kosinova podobnost

Kumpošt ve své disertační práci [15] využívá pro identifikaci počítačů metodu Kosinové podobnosti, popsanou v podkapitole 4.3.

Jako vstupní data měl k dispozici záznamy Netflow z univerzitní sítě z časového intervalu jeden měsíc. Pro označení sledovaných počítačů využil zdrojovou IP adresu počítačů v síti. Pro identifikaci využil atributy cílová IP adresa a počet navázaných spojení mezi počítači. Tyto atributy použil pro identifikaci v rámci služeb SSH na portu 22, HTTP na portu 80 a HTTPS na portu 443. Při zpracování vstupních dat využil transformaci typu inverse document frequency, vysvětlenou v podkapitole 3.3.

Úspěšnost identifikace pro službu SSH se pohybovala mezi 14.8% a 61.5%. Pro službu HTTP byla úspěšnost mezi 8% a 21% a pro službu HTTPS byla úspěšnost v rozmezí 11.8% až 26.5%.

4.5.3 Support Vector Machines

Panchenko ve své práci [18] využívá metodu Support Vector Machines pro identifikaci webových stránek, anonymizovaných pomocí síťě Tor.

Jako vstupní data využil odchycenou komunikaci se 4000 anonymizovanými webovými stránkami. Jako atributy pro identifikaci využil počet přenesených paketů, procento příchozích paketů, velikosti paketů, počet přenesených bajtů a značky velikosti datového toku popsané v podkapitole 3.2. Při využití všech zmíněných atributů dosáhl úspěšnosti identifikace 54.6%.

Kapitola 5

Návrh nástroje pro identifikaci počítačů

V této kapitole je popsán návrh programu pro identifikaci podle vzorů chování. Nejprve je vysvětlen výběr atributů určených pro identifikaci počítačů a je navržen typ vstupních dat. Ve druhé podkapitole je popsán protokol Netflow a komponenty sloužící k získávání záznamů Netflow. Také jsou vypsány atributy síťových toků, které jsou uložené v záznamech Netflow. Ve třetí podkapitole je vysvětleno vytváření modelu chování. Ve čtvrté podkapitole je navržen způsob identifikace počítačů. V poslední podkapitole jsou popsány krajní případy, které mohou při identifikaci počítačů nastat.

5.1 Výběr atributů

Při identifikaci počítačů pomocí vzorů síťového provozu se zpracovává velké množství dat. Při zpracování vstupních dat je snahá nalézt rovnováhu mezi přesnosti identifikace a objemem vstupních dat. Klíčová operace pro snížení množství dat je výběr atributů. Vhodný výběr atributů umožňuje snížit množství zpracovávaných dat a zároveň udržet velkou odlišitelnost uživatelů a tím i udržet vysokou úspěšnost klasifikace.

V této práci byly pro identifikaci počítačů vybrány následující atributy:

- Zdrojová IP adresa je použita pro značení jednotlivých počítačů. Také představuje výstupní hodnotu identifikace.
- Cílová IP adresa slouží k odlišení cílových stanic, se kterými sledovaný počítač komunikoval.
- Cílový port je vybrán pro rozpoznání služeb, které uživatel využívá. Filtrováním cílového portu lze se zaměřit na konkrétní služby, podle kterých se bude provádět identifikace.
- Počet přenesených paketů v rámci komunikace mezi sledovaným počitačem a jednou cílovou stanicí je vybrán jako atribut představující charakteristiku síťového toku. Pomocí počtu přenesených paketů lze odlišit sledovné počitače, komunikující se stejnou cílovou stanicí a využivající stejnou síťovou službu.

Na základě výše zmíněných atributů byly v této práci vybrány jako vstupní data záznamy Netflow síťového provozu, protože všechny vybrané atributy lze získat z těchto záznamů. Výhodou Netflow je již zpracovaný síťový provoz ve formě záznamu síťových toků.

To umožňuje rychlejši zpracování většího množství síťové komunikace, než tomu je při zpracování po paketech.

5.2 Netflow

Netflow je síťový protokol vyvinutý společností Cisco Systems. Slouží pro získávání informací o síťových tocích v rámci sítě. Netflow je dostupný v rámci proprietárního Cisco IOS, který je nainstalován na směrovačích a přepínačích této firmy. Protokol Netflow je uzavřený standart, ale existuje jeho specifikace v RFC 3954 [8]. Tato specifikace umožnila protokol Netflow implementovat také na jiných platformách. Protokol Netflow existuje v několika verzích. Informace získané z Netflow mohou být použity k různým učelům jako jsou například detekce anomálií nebo statistické monitorování.

5.2.1 Architektura Netflow

Netflow data jsou získána z datových toků počítačové sítě. Datový tok je v terminologii NetFlow definován jako sekvence paketů se shodnou zdrojovou IP adresou, cílovou IP adresou, zdrojovým portem, cílovým portem a číslem protokolu transportní vrstvy [1].

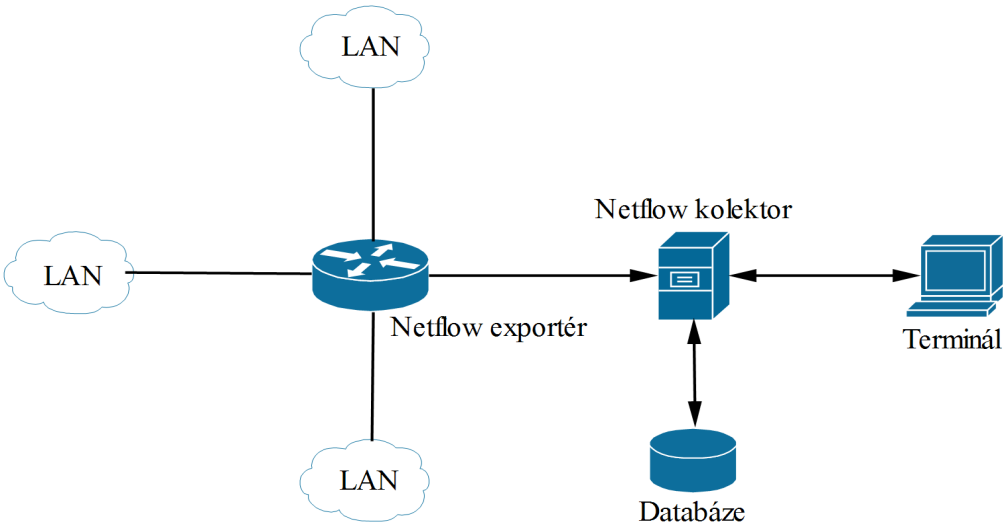
Pro získání a zpracování Netflow dat je potřeba několik komponent. První se nazývá Exportér. Exportér pasivně sleduje pakety a vytváří statistické záznamy o datových tocích v pozorované síti. Tyto záznamy jsou později odeslány do kolektoru pomocí protokolu Netflow. Exportéry jsou implementovány na aktivních síťových prvcích, to jsou směrovače nebo přepínače. V dnešní době se také používájí tzv. Netflow sondy. Jsou to samostatné zařízení, které se připojují do sítě a plní roli exportéra. Hlavní výhodou Netflow sondy je, že vytváření statistik již nezatěžuje výpočetní výkon aktivních síťových prvků. Další výhoda je pozorování a vytváření statistik ze všech paketů, oproti exportéru na aktivních síťových prvcích, kde bylo použito vzorkování.

Další komponentou architektury Netflow je kolektor. Netflow kolektor sbírá záznamy odeslané z jednoho nebo více exportéru a ukládá je na lokální úložiště nebo do databáze. Na obrázku 5.1 je příklad architektury s exportérem na směrovači.

5.2.2 Záznamy Netflow

Získané záznamy Netflow můžeme zobrazit a analyzovat například pomocí programu nfdump [3]. Ten zpracovává záznamy uložené v souborech ve formátu nfcapd. V následujícím seznamu jsou vypsány atributy síťových toků, které lze použít k analýze.

- **Start Time** - Čas prvního zaznamenaného paketu datového toku.
- **End Time** - Čas posledního zaznamenaného paketu datového toku.
- **Duration** - Délka toku v sekundách a milisekundách. Pokud jsou toky agregovány, je délka součtem délek v těchto tocích.
- **Protocol** - Protokol použitý v rámci spojení.
- **Source Address:Port** - Zdrojová IP adresa a zdrojový port.
- **Destination Address:Port** - Cílová IP adresa a cílový port.
- **Source AS** - Číslo zdrojového autonomního systému.



Obrázek 5.1: Ukázka architektury Netflow.

- **Destination AS** - Číslo cílového autonomního systému.
- **Input Interface num** - Zdrojové síťové rozhraní.
- **Output Interface num** - Cílové síťové rozhraní.
- **Packets** - Počet paketů v datovém toku. Pokud jsou datové toky agregovány, je počet paketů součtem počtu paketů v těchto tocích.
- **Bytes** - Počet přenesených bajtů v datovém toku.
- **Flows** - Počet toků v rámci záznamu. Pokud nejsou toky agregovány, hodnota je vždy jedna. Při agregaci představuje počet agregovaných toků.
- **TCP Flags** - Příznaky TCP paketů.
- **ToS** - Typ služby.
- **bps** - Počet bitů za sekundu.
- **pps** - Počet paketů za sekundu.
- **Bpp** - Počet bajtů na paket, vypočteno jako: počet bajtů / počet paketů.

5.3 Vytváření modelu chování

Při identifikaci počítačů se pracuje s trénovací a testovací množinou dat. Trénovací data obsahují komunikaci známých počítačů. Testovací data obsahují komunikaci počítačů, které

chceme identifikovat. Identifikace představuje přiřazení počítače z trénovací množiny k počítači z testovací množiny.

Trénovací a testovací množina dat se skládá z datových toků. Tyto datové toky jsou složeny z vybraných atributů, popsaných v podkapitole 5.1. Trénovací a testovací množiny dat je vhodné pro efektivnější práci strukturovat. Pro ukládání vstupních dat je vybrán vektorový model, popsaný v podkapitole 3.1. Při zpracování se vytváří dva vektorové modely. Referenční model z trénovacích dat a testovaný model z testovacích dat. V každém modelu se data ukládají do vektorů. Do jednoho vektoru se ukládají datové toky, sloučené podle společné *zdrojové IP adresy*, *cílové IP adresy* a *cílového portu*. Vektory se stejnou *zdrojovou IP adresou* představují chování jednoho počítače.

Součástí návrhu je také možnost zadat prefixy cílové IP adresy pro různé síťové služby dané cílovým portem. Datové toky jsou poté sloučeny podle adresy sítě, místo adresy konkrétního počítače. Příklad vytváření vektorového modelu je uveden na obrázku 5.2. Pro port 22 je zadaný prefix cílové IP adresy a při porovnání adres sítě jsou sloučeny datové toky dvou cílových počítačů.

Datové toky

zdrojová IP	cílová IP	cílový port	počet paketů
115.234.131.172	1.169.83.170	80	14
115.234.131.172	1.169.83.170	80	27
115.234.131.172	15.235.157.113	443	32
115.232.136.248	146.37.132.106 /24	22	11
115.232.136.248	146.37.132.112 /24	22	42

Vektory

zdrojová IP	cílová IP	cílový port	počet paketů
115.234.131.172	1.169.83.170	80	41
115.234.131.172	15.235.157.113	443	32
115.232.136.248	146.37.132.0	22	53

Počítače

zdrojová IP
115.234.131.172
115.232.136.248

Obrázek 5.2: Příklad zpracování datových toků.

Jelikož trénovací a testovací záznamy Netflow mohou být z různě dlouhého časového úseku, je atribut *počtu přenesených paketů* normalizován. Normalizace se provádí podle počtu toků a aplikuje se na všechny vektory v referenčním a testovaném vektorovém modelu. Normalizace atributu *počtu přenesených paketů* se provádí pomocí vzorce 5.1:

$$n_{p,v} = \frac{f_{p,v}}{t_v} \tag{5.1}$$

Proměnná t_v představuje počet toků, ze kterých byl vytvořen vektor v. $f_{p,v}$ představuje

hodnotu atributu *počtu přenesených paketů* ve vektoru v. Proměnná $n_{p,v}$ je normovaná hodnota atributu *počtu přenesených paketů*.

V této práci je také využita transformace typu *term frequency*, popsaná v podkapitole 3.3. Transformace typu *term frequency* se aplikuje u všech vektorů v referenčním vektorovém modelu a testovaném vektorovém modelu na atribut *počtu přenesených paketů*.

5.4 Průběh identifikace počítače

Identifikaci počítačů lze převést na klasifikační problém z oblasti dolování dat [11]. Vybrané metody pro dolování dat, které se používají k řešení klasifikačních problémů a zároveň se dají použít k identifikaci počítačů jsou popsány v kapitole 4. V této práci je proces identifikace navržen podle metody Multinomial Naive Bayes, popsané v podkapitole 4.1.

Při identifikaci se porovnávají všechny vektory dvou počítačů, jednoho počítače z vektorového modelu vytvořeného z trénovacích dat a druhého počítače z vektorového modelu vytvořeného z testovacích dat.

Dva vektory jsou nejdříve porovnány na shodu *cílové IP adresy* a *cílového portu*. Pro dva porovnávané vektory se stejnou *cílovou IP adresou* a stejným *cílovým portem* se vypočte podmíněná pravděpodobnost. V této práci se nepočítá podmíněná pravděpodobnost ze samotných atributů, ale z jejich poměrů. Poměrová hodnota atributu *počtu přenesených paketů* $r_{p,v}$ pro vektor v počítače z referenčního modelu je vypočtena pomocí vzorce 5.2:

$$r_{p,v} = \begin{cases} \dfrac{n_{p,w}}{n_{p,v}} & \text{pokud } n_{p,v} \geq n_{p,w} \\ \dfrac{n_{p,v}}{n_{p,w}} & \text{pokud } n_{p,v} < n_{p,w} \end{cases} \tag{5.2}$$

Také je spočtena poměrová hodnota atributu *počtu přenesených paketů* $r_{p,w}$ pro vektor w počítače z testovaného modelu. Ten je vypočten pomocí vzorce 5.3:

$$r_{p,w} = \begin{cases} \dfrac{n_{p,v}}{n_{p,w}} & \text{pokud } n_{p,v} \geq n_{p,w} \\ \dfrac{n_{p,w}}{n_{p,v}} & \text{pokud } n_{p,v} < n_{p,w} \end{cases} \tag{5.3}$$

Proměnná $n_{p,v}$ je atribut *počet přenesených paketů* z vektoru v počítače nacházejícího se v referenčním modelu. $n_{p,w}$ je atribut *počet přenesených paketů* z vektoru w počítače v testovaném modelu.

Z výše vypočtených poměrů se vypočte podmíněná pravděpodobnost pomocí vzorce 5.4:

$$P(v_p|c) = \frac{r_{p,v} + 1}{N_p + V} \tag{5.4}$$

$P(v_p|c)$ je podmíněná pravděpodobnost výskytů vektorů v_p u referenčního počítače c, určena na základě poměru atributů *počtu přenesených paketů*. Proměnná $r_{p,v}$ představuje poměr atributů *počtu přenesených paketů*, vypočteného ve vzorci 5.2. Proměnná N_p je suma *počtu přenesených paketů* v rámci počítače z referenčního modelu. Proměnná V představuje počet všech vektorů z referenčního modelu.

Poté celkovou míru pravděpodobnosti $P(c|k)$, že testovaný počítač k je shodný s počítačem c z referenčního modelu získáme pomocí vzorce 5.5:

$$P(c|k) \approx \prod_{h \in H} P(v_p|c)^{r_{p,w}} \tag{5.5}$$

Proměnná $r_{p,w}$ představuje poměrovou hodnotu atributů *počtu přenesených paketů*, získanou ze vzorce 5.3.

Výpočet míry pravděpodobnosti popsaný výše je proveden, pokud je nalezena shoda *cílové IP adresy* a *cílového portu*. Pokud se vektor testovaného počítače s danou *cílovou IP adresou* a daným *cílovým portem* nenachází u porovnávaného počítače z referenčního modelu, je provedena penalizace. Penalizace se provádí pomocí maximální a minimální hodnoty atributu *počtu přenesených paketů* z referenčního a testovaného vektorového modelu. Na základě těchto hodnot jsou vypočteny poměrové hodnoty dle vzorců 5.6 a 5.7:

$$r_{p,min} = \frac{n_{p,min}}{n_{p,max}} \tag{5.6}$$

$$r_{p,max} = \frac{n_{p,max}}{n_{p,min}} \tag{5.7}$$

Proměnná $n_{p,min}$ je minimální a proměnná $n_{p,max}$ je maximální hodnota atributu *počtu přenesených paketů* z obou vektorových modelů. $r_{p,min}$ je minimální poměrová hodnota a $r_{p,max}$ je maximální poměrová hodnota. Poté je spočtena penalizace dle vzorce 5.8:

$$p_{v,c} = \left(\frac{r_{p,min} + 1}{N_p + V} \right)^{r_{p,max}} \tag{5.8}$$

Proměnná $p_{v,c}$ je penalizace testovaného počítače, že se vektor v nevyskytuje u referenčního počítače c. Proměnná $r_{p,min}$ je minimální poměrová hodnota vypočtená ve vzorci 5.6. $r_{p,max}$ je maximální poměrová hodnota vypočtena ve vzorci 5.7. N_p je suma atributu *počtu přenesených paketů* u referenčního počítače. Proměnná V představuje počet všech vektorů z referenčního modelu. Penalizace je posléze přinásobena k celkové míře pravděpodobnosti shody $P(c|k)$ dle vzorce 5.9:

$$P(c|k) = P(c|k) \cdot p_{v,c} \tag{5.9}$$

U jednoho testovaného počítače je proveden výpočet celkové míry pravděpodobnosti pro všechny počítače z referenčního modelu. Následná identifikace je provedena přidělením referenčního počítače s největší celkovou mírou pravděpodobnosti k danému testovanému počítači.

5.5 Krajní případy při identifikaci počítačů

V této podkapitole jsou popsány některé události, které mohou nastat při identifikaci počítačů pomocí navrženého nástoje.

5.5.1 Přiřazení referenčního počítače k více testovaným počítačům

V rámci identifikace se provádí přiřazování referenčního počítače k testovanému počítači na základě největší míry pravděpodobnosti shody počítačů. Navržený nástroj nekontroluje opakované přiřazení stejného referenčního počítače k více testovaným počítačům. Tato vlastnost pokrývá situaci, kdy je v testované množině jeden počítač reprezentovaný více zdrojovými IP adresami.

5.5.2 Testovaný počítač nelze nalézt v referenčním modelu

Při identifikaci počítačů může nastat případ, kdy pro testovaný počítač nebudou existovat záznamy komunikace v trénovacích datech. V tomto případě nebude k testovanému počítači existovat patřičný referenční počítač a tedy nebude možné tento testovaný počítač identifikovat.

Navržený nástroj vždy přiřadí referenční počítač na základě nejvyšší vypočtené míry pravděpodobnosti shody bez ohledu na to, jestli je nejvyšší míra pravděpodobnosti nižší než u jiných, úspěšně identifikovaných testovaných počítačů. Tento případ lze ošetřit nastavením prahové hodnoty míry pravděpodobnosti. Pokud nejvyšší míra pravděpodobnosti shody je nižší než prahová hodnota, nebude k testovanému počítači přiřazen žádný referenční počítač. Pro výpočet a nastavení prahové hodnoty jsou provedeny experimenty v sekci 7.2.2.

5.5.3 Odstranění cílové IP adresy z identifikace

Navržený nástroj umožňuje v rámci cílového portu zadat prefix /0 pro cílovou IP adresu. To znamená, že všechny cílové IP adresy budou mít stejnou hodnotu a při identifikaci bude porovnání cílových IP adres na shodu vždy pravdivé. Tato vlastnost umožňuje pro identifikaci počítačů využít i služby, při jejichž využívání se dynamicky komunikuje s různými cílovými servery s různými IP adresami.

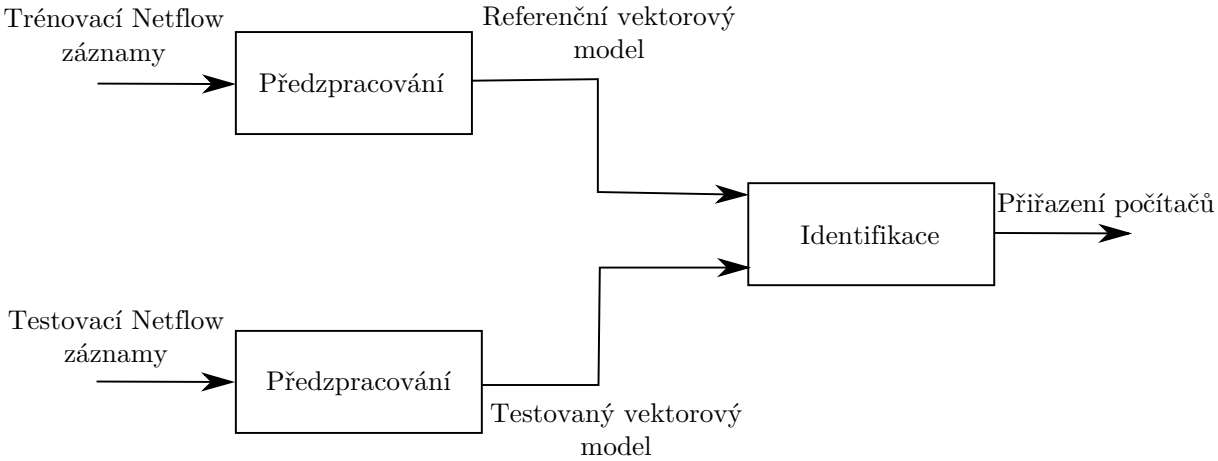
Kapitola 6

Implementace

V této kapitole je popsána implementace nástroje pro identifikaci počítačů. V první podkapitole je popsáno schéma nástroje a rozdělení nástroje do modulů. V modulu předzpracování probíhá zpracování záznamů Netflow. Detailnější popis zpracování vstupních dat je popsán v podkapitole 6.2. Součásti předzpracování je také vytváření vektorových modelu a jejich ukládání do souboru ve formátu XML. Bližší popis je vysvětlen v podkapitole 6.3. Program je implementovaný v jazycích C++ a Bash.

6.1 Popis programu

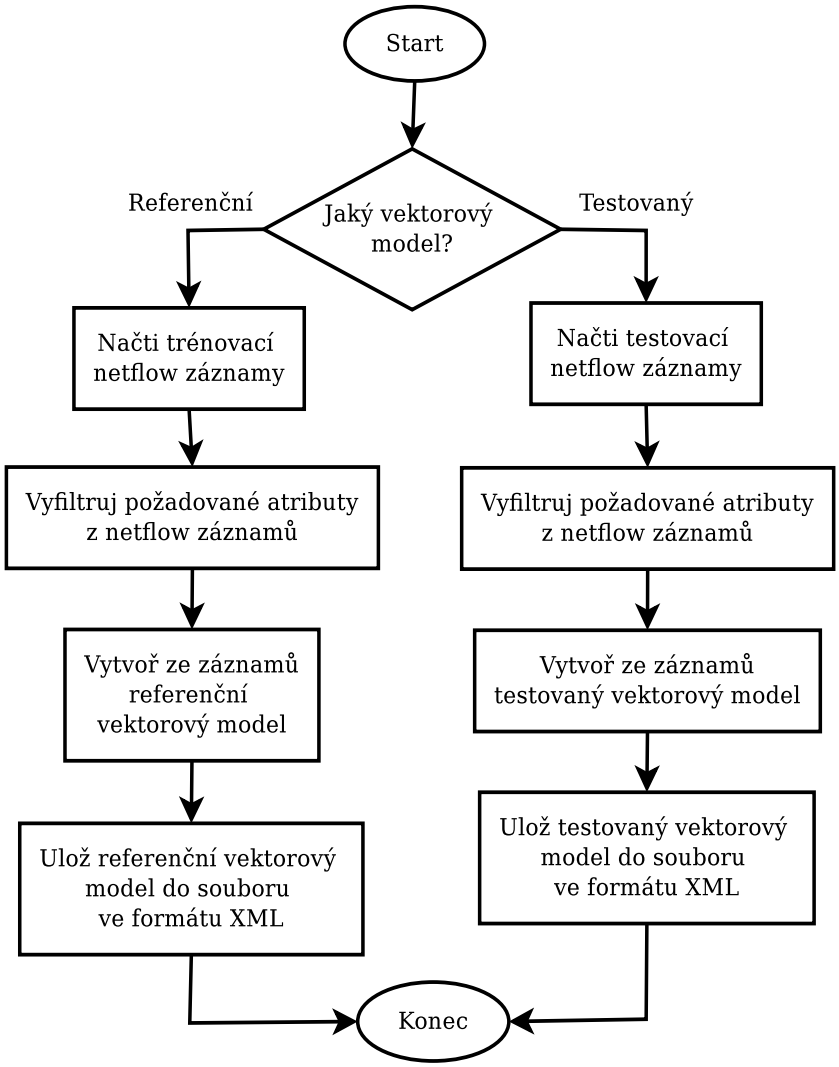
Navržený program pro identifikaci počítačů je rozdělen do dvou modulů. Toto rozdělení je znázorněno na obrázku 6.1. V rámci předzpracování se vytváří referenční vektorový model ze zpracovaných trénovacích záznamů Netflow. Při předzpracování se také z testovacích záznamů Netflow vytváří testovaný vektorový model. V modulu identifikace se provádí vzájemné přiřazování počítačů z vektorových modelů.



Obrázek 6.1: Schéma rozdělení programu do modulů.

Vývojový diagram modulu předzpracování je zobrazen na obrázku 6.2. Při předzpracování se nejdřív rozhodne, zda mají být zpracovány testovací nebo trénovací záznamy Netflow. Podrobnější popis zpracování vstupních dat je popsán v podkapitole 6.2. Dle výběru se vytváří buď referenční vektorový model z trénovacích záznamů Netflow nebo testovaný vek-

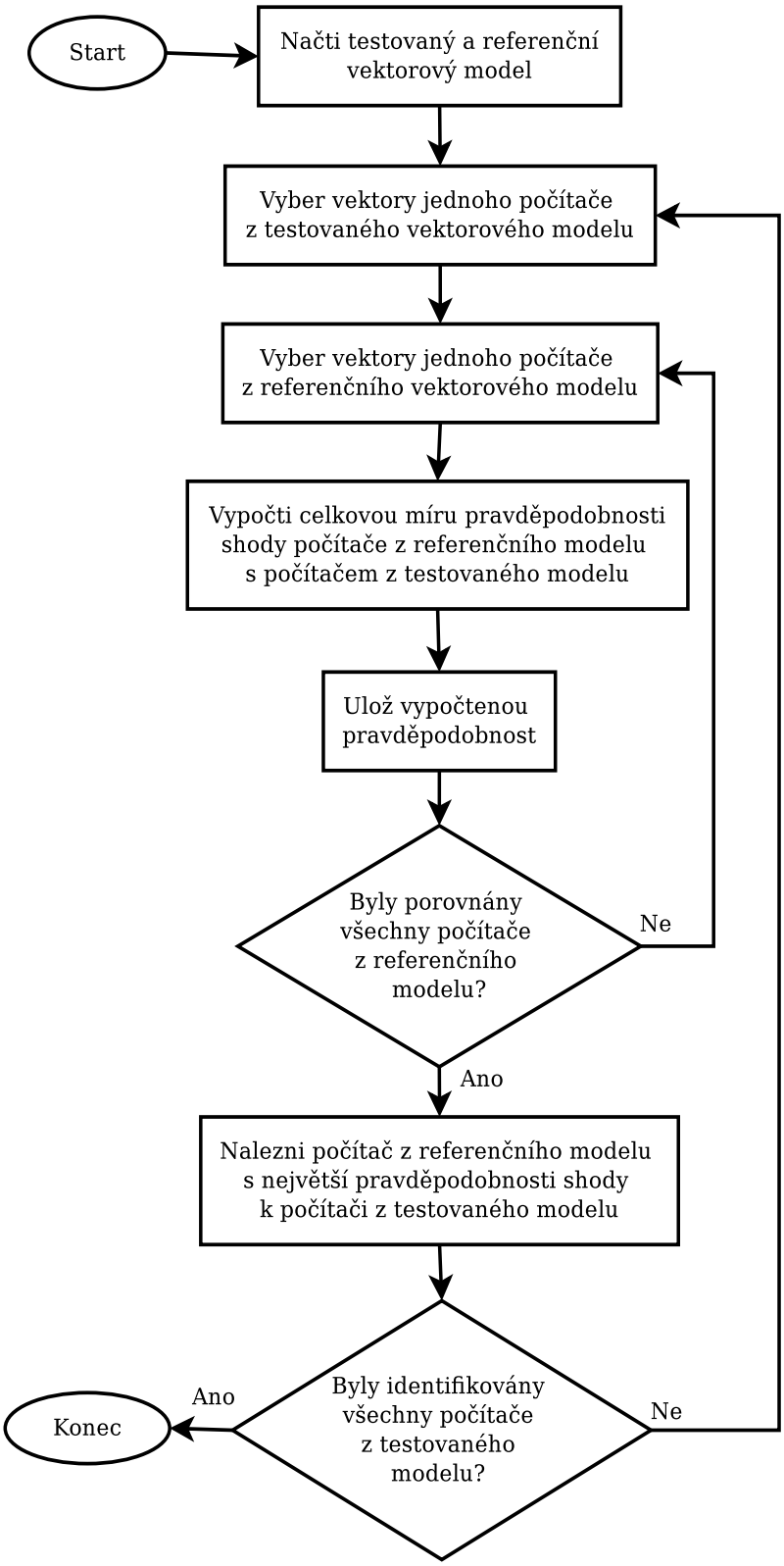
torový model z testovacích záznamů Netflow. Oba vektorové modely se ukládají do souboru ve formátu XML.



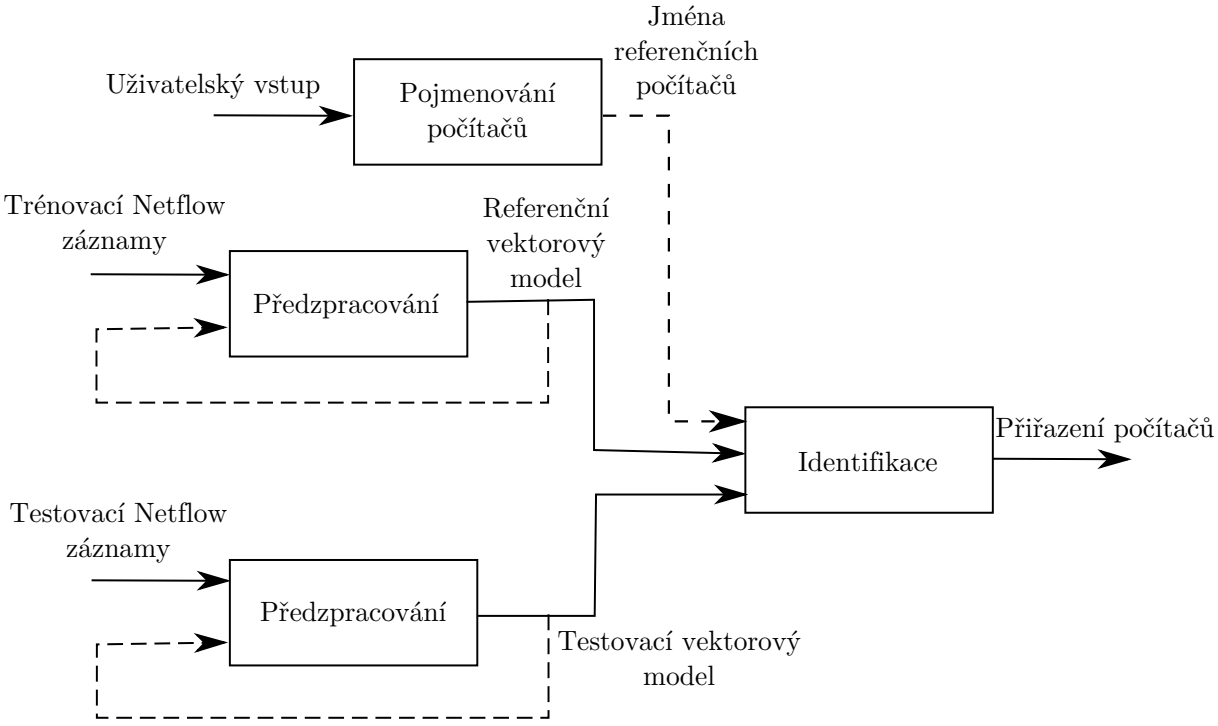
Obrázek 6.2: Vývojový diagram modulu předzpracování vstupních dat.

Vývojový diagram modulu identifikace je zobrazen na obrázku 6.3. V tomto modulu jsou načteny oba vektorové modely z XML souborů. Poté jsou postupně vybírány počítače z testovaného modelu. Pro každý počítač z testovaného modelu je vypočtena míra pravděpodobnosti shody pro všechny počítače z referenčního modelu. Výpočet míry pravděpodobnosti shody je popsán v podkapitole 5.4. Při identifikaci je počítač z testovaného modelu přidružen k počítači z referenčního modelu s nejvyšší mírou pravděpodobnosti shody.

Na obrázku 6.4 je zobrazeno schéma se všemi možnostmi programu. Čárkované šipky znázorňují volitelnou funkčnost programu. Zpětné vazby u modulu předzpracování představují aktualizaci vektorových modelů, popsanou v podkapitole 6.3. V modulu pojmenování počítače může uživatel přidat jména k referenčním počítačům. Tato možnost je blíže popsána v podkapitole 6.3.



Obrázek 6.3: Vývojový diagram modulu pro identifikaci počítačů.



Obrázek 6.4: Celkové schéma nástroje pro identifikaci počítačů.

6.2 Zpracování vstupních dat

Jako vstupní data byly vybrány Netflow záznamy ve formátu *nfcap*. Pro zpracování záznamů Netflow je využit program *nfdump*[1]. Ten slouži k převodu Netflow záznamu do textového formátu a filtrování požadovaných atributů k identifikaci. Také může sloužit pro získávání statistik ze vstupních dat. Dokaže provádět agregaci síťových toků podle zadaných atributů, které mohou být například *zdrojová* nebo *cílová IP adresa*. *Nfdump* podporuje záznamy Netflow ve verzích 5, 7 a 9.

Ukázka využití programu *nfdump* k filtrování požadovaných atributů z vybraného časového úseku:

nfdump -A *srcip,dstip,srcport,dstport* -R */netflow/první:poslední* -o "fmt:%sap %dap %pkt %fl" "*src or dst port in* [22 80]"

- **Přepínač -A** - Provádí agregaci podle zadaných atributů.
- **Přepínač -R** - Čte postupně zadanou sekvenci od prvního zadaného souboru po poslední.
- **Přepínač -o** - Provádí formátování výstupů programu.
- **Parametr port in** - Filtruje záznamy podle zadaného seznamu portů.

Program *nfdump* v rámci síťové komunikace určuje *zdrojovou IP adresu* podle odesílatele paketu a *cílovou IP adresu* podle příjemce paketu. V této práci je *zdrojová IP adresa*

[1] http://nfdump.sourceforge.net

přiřazena klientovi, který zahájil komunikaci při využití určené služby. Rozlišení IP adresy, patřící počítači, který zahájil komunikaci se provádí pomocí *cílového portu*.

Čísla portů se zadávájí pro filtraci služeb, podle kterých chceme identifikovat počítače. Port u načteného síťového toku, který se shoduje s některým ze zadaných portů je označen jako *cílový port*. *Cílová IP adresa* je tudíž adresa počítače, který skrze tento port komunikoval. Pokud se u síťového toku oba porty shodují s některým ze zadadných portu, není potom možné určit *zdrojovou* a *cílovou IP adresu* a daný záznam je ignorován.

6.3 Operace s vektorovými modely

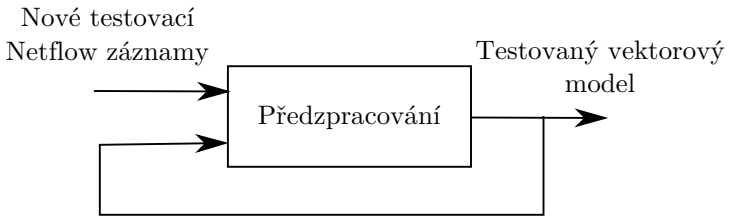
V podkapitole 5.3 je popsáno vytváření vektorového modelu. Vytvořené vektorové modely jsou ukládány do souboru ve formátu XML. Pro vytváření a čtení XML souboru je využita knihovna *TinyXML-2*[2].

Pro zadávání názvů XML souborů je vytvořen konfigurační soubor. Do konfiguračního souboru se vkládá cesta k XML souboru, do kterého má být uložen referenční vektorový model a cesta k XML souboru, do kterého má být uložen testovaný vektorový model. Tyto soubory jsou vytvořeny při předzpracování a posléze načteny při identifikaci, jak je to znázorněno na obrázku 6.1.

Program umožňuje aktualizaci referenčního vektorového modelu s použitím nových trénovacích záznamů Netflow, jak je zobrazeno na obrázku 6.5. Pro tuto aktualizaci se přidává do konfiguračního souboru volitelná položka s názvem původního XML souboru referenčního vektrového modelu.

Obrázek 6.5: Schéma aktualizace referenčního vektorového modelu.

Program také umožňuje aktualizaci testovaného vektorového modelu s použitím nových testovacích záznamů Netflow. Princip aktualizace je demonstrován na obrázku 6.6. Pro tuto aktualizaci se přidává do konfiguračního souboru volitelná položka s názvem původního XML souboru testovaného vektrového modelu.



Obrázek 6.6: Schéma aktualizace testovaného vektorového modelu.

Pro pojmenování počítačů z referenčního vektorového modelu se vytváří XML soubor se *zdrojovými IP adresami*, ke kterým lze přidat název počítače. Pro možnost pojmenování

[2] http://www.grinninglizard.com/tinyxml2/index.html

počítačů se zadává do konfiguračního souboru položka s názvem XML souboru, obsahující dvojice název počítače a *zdrojová IP adresa* počítače z referenčního vektorového modelu.

Kapitola 7

Experimenty

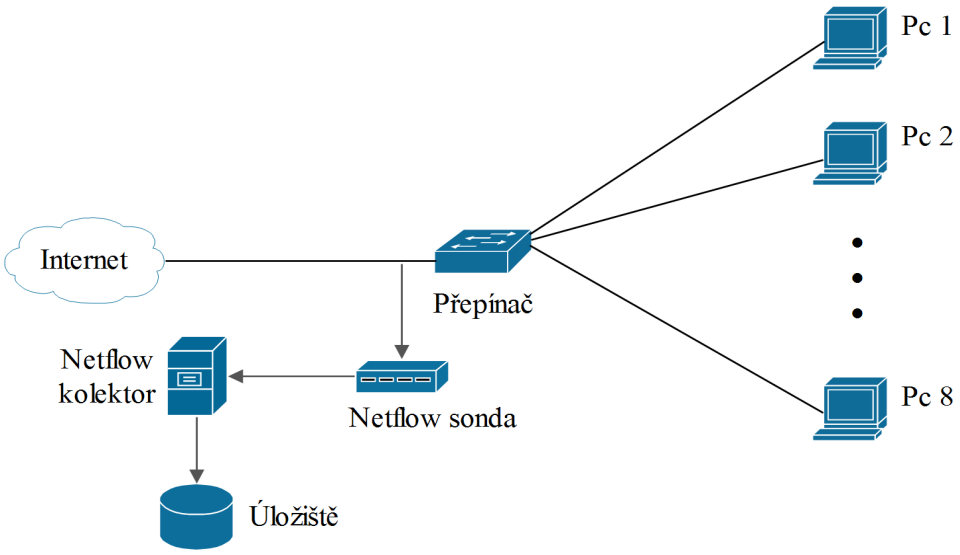
V této kapitole jsou popsány provedené experimenty za účelem zjištění úspěšnosti identifikace počítačů pomocí navrženého nástroje. Experimenty jsou rozděleny na dvě časti podle použitých vstupních dat. V první části jsou provedeny experimenty na simulovaných datech. V druhé části jsou provedeny experimenty na datech z reálného prostředí. V rámci experimentů s reálnými daty jsou také provedeny experimenty na zjištění doby zpracování dat a identifikace počítačů. Všechny experimenty jsou provedeny na počítači s procesorem Intel Core2 Duo 2GHz, operační paměti o velikosti 2,5GB a operačním systémem Ubuntu 12.10.

7.1 Experimenty s daty ze simulovaného prosředí

Pro získání dat je vytvořena virtuální síť v prostředí *Mininet*[1]. Architektura virtuální sítě je znázorněna na obrázku 7.1. V síti je osm počítačů komunikujících s různými cílovými stanicemi v internetu. Jako Netflow sonda je využit nástroj *fprobe*[2], který sleduje provoz mezi prostředím *Mininet* a internetem. *Fprobe* vytváří statistiky o datových tocích a posílá je Netflow kolektoru. Jako Netflow kolektor je použit nástroj *nfcapd*[3]. V tomto experimentu jsou pro generování síťového provozu využity nástroje *iperf*[4] a *wget*[5]. Síťový provoz je generován pro služby HTTP na portu 80 a HTTPS na portu 443. Chování počítačů je simulováno pomocí skriptů implementovaných ve skriptovacím jazyce Bash. Každý počítač komunikuje celkem s deseti webovými servery. Počítače navazovaly spojení s webovými servery každých šest hodin.

Pro vytvoření referenčního modelu jsou použity záznamy Netflow z jednoho dne odchycené komunikace. Jako testovací data jsou použity záznamy z následujícího dne. Pro základní ověření úspěšnosti identifikace počítačů je nastaveno pro všechny počítače sedm webových serverů společných a tři rozdílné. Toto nastavení je zachováno pro oba použité dny. Identifikace na základě těchto dat je pro všechny testované počítače úspěšná. Na obrázku 7.2 je zobrazen graf míry pravděpodobnosti shody testovaného počítače $Pc1$ se všemi referenčními počítači. V grafu lze vidět, že stejný počítač $Pc1$ je jednoduché odlišit od ostatních. Míra pravděpodobnosti shody stejného počítače je přibližně o patnáct řad vyšší než u druhého počítače $Pc8$. Dobrá odlišitelnost závisí na atributech vybraných pro identifikaci

[1] http://mininet.org/
[2] http://fprobe.sourceforge.net/
[3] http://nfdump.sourceforge.net/
[4] http://iperf.fr/
[5] https://www.gnu.org/software/wget/



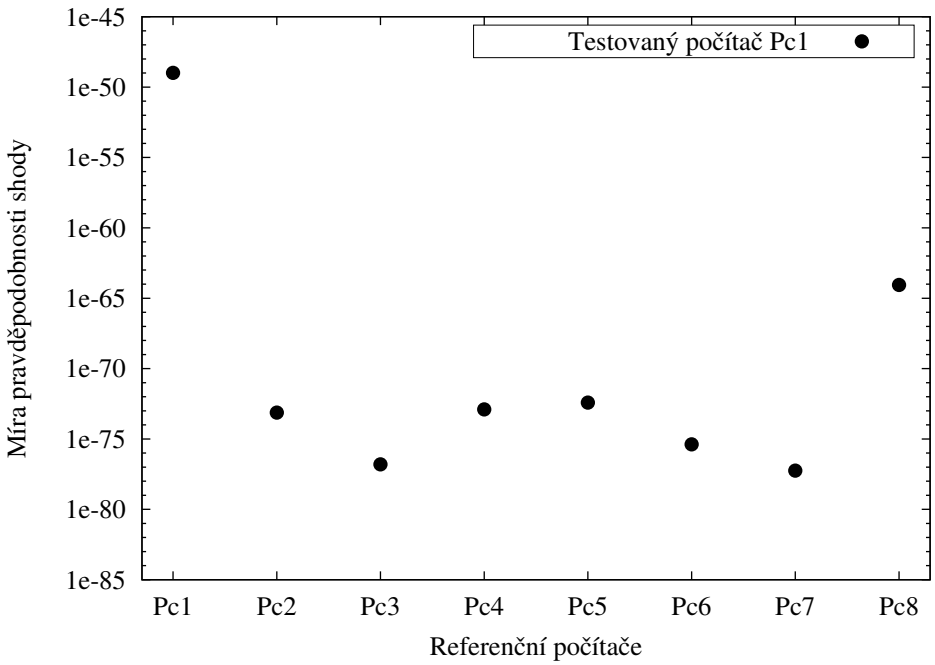
Obrázek 7.1: Schéma vytvořené sítě v prostředí Mininet.

počítačů. S vytvořeným nástrojem se nejlépe odlišují počítače, které komunikují s různými cílovými stanicemi a s různým počtem přenesených paketů v rámci navázaného spojení než ostatní počítače.

Pro zjištění, za jakých okolností je navržený nástroj schopný úspěšně identifikovat počítač jsou postupně měněny webové servery, se kterými počítače komunikovaly. Tím se také postupně snižovala odlišitelnost referenčních počítačů. K neúspěšné identifikaci počítače docházelo z několika důvodů. V prvním případě došlo k neúspěšné identifikaci, když se změnily webové servery tak, že testovaný počítač měl méně společných vektorů se svým referenčním vzorem, než s jiným referenčním počítačem. Společny vektor představuje vektor s danou cílovou IP adresou a daným cílovým portem a tento vektor se vyskytuje v referenčním i testovaném počítači.

K neúspěšné identifikaci občas docházelo, když testovaný počítač měl stejný počet společných vektorů se svým referenčním vzorem a jiným referenčním počítačem. Při stejném počtu shodných vektorů o shodě počítačů rozhoduje atribut *počtu přenesených paketů*. V tomto případě způsobovala neúspěšnou identifikaci hodnota sumy počtu přenesených paketů v rámci referenčního počítače. Ta je použita pro výpočet podmíněné pravděpodobnosti dle vzorce 5.4, ve kterém je označena jako proměnná N_p.

Tento případ vysvětlím pro vektor z testovaného počítače, který se nachází ve dvou různých referenčních počítačích a má stejnou hodnotu atributu *počtu přenesených paketů* ve všech třech počítačích. Pokud je hodnota sumy u jednoho referenčního počítače nižší než u druhého, pak je podmíněná pravděpodobnost vektorů vyšší, i když je hodnota atributu *počtu přenesených paketů* stejná. Tímto jsou ovlivněny i ostatní vektory a tento případ může vést k neúspěšné identifikaci.



Obrázek 7.2: Graf míry pravděpodobnosti shody testovaného počítače Pc1 se všemi referenčními počítači.

7.2 Experimenty s daty z reálného prosředí

Jako data z reálného prostředí jsou vybrány anonymizované Netflow záznamy ze sítě VUT. K dispozici jsou záznamy ze dvou po sobě následujících dnů. Pro experimentální měření úspěšnosti je potřeba znát vztahy mezi referenčními a testovanými počítači. To lze zajistit pomocí trvale stejné IP adresy pro každý sledovaný počitač. U protokolu IPv6 se může adresa dynamicky měnit a není jisté, že IPv6 adresa jednoho počítače bude v obou dnech stejná. Proto jsou všechny experimenty provedený pouze s IPv4 adresami.

Základní rozdíl těchto dat oproti simulovaným je případ, kdy se komunikace jednoho počítače nenachází v obou vstupních množinách dat. Tento problém je popsán v sekci 5.5.2. Další zjištěná komplikace je velký počet počítačů s malým počtem navázaných spojení ve zpracovaném časovém úseku. Častým případem je počítač pouze s jedním navázavým spojením. Tento případ značně ztěžuje identifikaci počítačů, protože z malého počtu vektorů nelze počítače jednoznačně odlišit. Velký počet počítačů také prodlužuje dobu trvání identifikace. Z tohoto důvodů je při experimentech přidáno filtrování referenčních a testovaných počítačů na určity počet vektorů. Pro všechny experimenty je nastaven filtr na minimálně tři vektory u referenčních i testovaných počítačů.

7.2.1 Měření úspěšnosti pro různé porty

Pro provedený experiment jsou z každého dne vybrány hodinové záznamy komunikace. Hodinový záznam z prvního dne je použitý pro vytvoření referenčního modelu a z hodinového záznamu z druhého dne je vytvořen testovaný model. V tabulce 7.4 jsou popsány základní parametry hodinových intervalů Netflow záznamů získaných ze dvou dnů.

Experiment je provedený nejen pro samostatné porty, ale i s využitím více portů na-

Název	Počet datových toků	Počet bajtů	Počet paketů
2014-01-28	24 358 489	850 GB	1 mld.
2014-01-29	26 630 041	798 GB	933 mil.

Tabulka 7.1: Základní charakteristiky vybraných hodinových intervalů Netflow záznamů ze sítě VUT.

jednou. Při zadání více portů najednou jsou pro identifikaci použity referenční a testované počítače, které komunikují skrze všechny zadané porty. V rámci experimentu je také využito prahování pro určení, zda se testovaný počítač nachází v referenčním modelu. Výpočet prahové hodnoty je blíže specifikován v podsekci 7.2.2. Prahový parametr je u tohoto experimentu nastaven na hodnotu 0,9. V tabulce 7.2 jsou vypsány úspěšnosti identifikace počítačů pro různé zadané porty nebo jejich kombinace.

Služby	Referenční	Testované	Společné	Úspěšná pozitivní	Úspěšná negativní
HTTP	6076	6416	3874	340	957
HTTPS	5131	5026	3092	282	568
SSH	147	146	121	10	19
SNMP	18	15	9	4	3
HTTP, HTTPS	5472	6017	3539	194	1195
HTTP, SSH	60	34	17	6	14
HTTPS, SSH	58	62	27	13	16

Tabulka 7.2: Úspěšnost identifikace počítačů pro různé služby a jejich kombinace.

V prvním sloupci jsou vypsány jednotlivé služby, v rámci kterých je provedena identifikace počítačů. Ve druhém sloupci je uveden počet počítačů v referenčním modelu. Ve třetím sloupci je počet počítačů v testovaném modelu. Ve sloupci *Společné* je uveden počet stejných počítačů, nacházejících se v obou vektorových modelech. Ve sloupci *Úspěšná pozitivní* se nachází počet úspěšně identifikovaných stejných počítačů. V posledním sloupci je uvedena úspěšná negativní identifikace počítačů. To znamená počet počítačů, které jsou správně označené jako neidentifikovatelné, při zadaném prahovém parametru o hodnotě $0,9$.

Nejlepší úspěšná pozitivní identifikace počítačů je u kombinace služeb HTTPS a SSH. U této kombinace je úspěšně identifikováno 48% testovaných počítačů z počítačů, které se vyskytují v obou vektorových modelech. Nejlepší celková úspěšnost identifikace je u kombinace služeb HTTP a SSH, u které je ze všech testovaných počítačů 58% správně identifikováno nebo správně označeno jako neidentifikovatelné.

7.2.2 Určení prahu pro stanovení neúspěšného nalezení počítače

V sekci 5.5.2 je popsána problematika, kdy nelze úspešně identifikovat testovaný počítač. Pro tento případ je vytvořena prahová hodnota, kterou je možné parametrizovat. Pro získání prahové hodnoty T jsem navrhl vzorec 7.1:

$$T = (p_{v,c})^{B \cdot S_k} \tag{7.1}$$

Proměnná $p_{v,c}$ je penalizace vypočtená ve vzorci 5.8. S_k je počet vektorů v testovaném počítači k. Proměnná B je vstupní parametr prahování, který určuje jak moc mohou být počítače odlišné. Parametr B je v rozmezí hodnot 0 až 1, kdy hodnota 0 značí, že počítače musí mít naprosto stejné chování, aby nebyly označené jako neidentifikovatelné. Prahový parametr o hodnotě 1 značí, že počítače mohou mít naprosto odlišné chování.

V rámci tohoto experimentu jsou jako trénovací vstupní data použité Netflow záznamy z desetiminutového intervalu z prvního dne. Jako testovací data jsou použité záznamy z desetiminutového intervalu ze druhého dne. Záznamy byly filtrovány podle služby HTTPS na portu 443. V referenčním modelu je uloženo celkem 2456 počítačů a testovaném modelu je uloženo celkem 2308 počítačů. V obou modelech se dohromady vyskytuje 1124 stejných počítačů. A tedy 1184 počítačů v testovaném modelu nebude možné identifikovat. V tabulce 7.4 jsou zobrazeny úspěšnosti identifikace pro různé hodnoty parametru prahování.

Prahový parametr	Úspěšná pozitivní	Úspěšná negativní
1	223	0
0,9	221	234
0,8	215	411
0,7	201	575
0,6	170	757
0,5	130	840
0,4	106	876
0,3	67	990
0,2	32	1075
0,1	30	1119

Tabulka 7.3: Úspěšnost identifikace počítačů pro různé hodnoty prahového parametru.

V prvním sloupci jsou uvedeny parametry prahové funkce popsané ve vzorci 7.1. V následujícím slouci se nachází úspěšná pozititní identifikace počítačů. To znamená počet úspěšně přiřazených stejných počítačů z referenčního a testovaného modelu. Ve třetím sloupci je úspěšná negativní identifikace počítačů. Ta představuje počet správně označených testovaných počítačů, pro které neexistuje stejný referenční počítač. Testovaný počítač je označen jako neidentifikovatelný, pokud prahová hodnota vypočtená pro daný testovaný počítač je větší než nejvyšší hodnota míry pravděpodobnosti shody vypočtená pro všechny referenční počítače.

V prvním řádku tabulky 7.4 je provedena identifikace s prahovým parametrem o hodnotě 1. Při této hodnotě jsou všechny testované počítače označeny jako identifikovatelné. Se snižující se hodnotou parametru prahování se zvyšuje hodnota vypočteného prahu. To znamená testovaný počítač musí být více shodný s referenčním počítačem, aby nebyl označen jako neidentifikovatelný. Tím také vzniká pokles úspěšné pozitivní identifikace počítačů. To je způsobeno tím, že úspěšně identifikované testované počítače jsou označovány jako neidentifikovatelné z důvodu příliš malé shody s počítačem v referenčním modelu.

7.2.3 Doba trvání běhu programu

V této části je měřena doba trvání programu na základě délky časového intervalu záznamů Netflow. Pro měření doby trvání běhu jednotlivých modulů je využit nástroj *time*. Jako výsledná hodnota doby trvání je využita hodnota *user*, která představuje dobu strávenou

prováděním programu. V tabulce 7.4 je zobrazena doba trvání předzpracování a identifikace pro různě dlouhé časové intervaly záznamů Netflow. Ve sloupci *Použité vektory* je počet vektorů, které jsou využity při identifikaci. Tento výběr je proveden na základě filtru minimálního počtu vektorů u počítačů, popsaného v úvodu této kapitoly.

Interval	Počet toků	Počet vektorů	Použité vektory	Předzpracování	Identifikace
10m	803 880	144 984	63 490	14s	58s
20m	1 567 708	267 567	100 753	30s	2m11s
60m	4 418 032	690 075	231 179	1m29s	8m15s
120m	8 738 206	1 230 567	398 826	3m37s	24m57s

Tabulka 7.4: Doba trvání běhu modulů pro různé časové intervaly Netflow záznamů.

Do doby předzpracování je zahrnuta doba vytváření referenčního i testovaného modelu. Z vybraného časového intervalu je polovina záznamů Netflow použita jako trénovací data a druhá polovina jako testovací data. V rámci předzpracování se také používá program *nfdump* pro filtraci záznamů a jejich převod do textové podoby. Doba běhu programu *nfdump* z vypsaných dob trvání předzpracování představovala průměrně 36%. Pro filtrování záznamů jsou použity port 80 pro službu HTTP a port 22 pro službu SSH. V získaných časových intervalech je u služby HTTP daleko větší síťový provoz než je u služby SSH. Z celkového počtu toků představovala služba SSH průměrně jen 9%. Oproti tomu doba předzpracování jen pro službu SSH je průměrně 19%.

7.3 Zhodnocení výsledků

Při experimentech s daty z reálného prostředí se objevovalo velké množství počítačů, které se vyskytovaly pouze v testovaném modelu. Proto byla vytvořena prahová funkce pro označení těchto počítačů. U prahování při každém zadaném parametru menším než jedna došlo k označení úspěšně identifikovaného počítače jako neidentifovatelný. Nejméně takto nesprávně označených počítačů bylo u prahového parametru o hodnotě 0,9. Pro tuto hodnotu byly 2 počítače z 223 úspěšně identifikovaných počítačů nesprávně označené jako neidentifikovatelné. Zároveň 234 počítačů z 1184 počítačů vyskytujících se pouze v testovaném modelu bylo správně označeno jako neidentifikovatelné.

V rámci experimentů, u kterých se měřila úspěšnost pro vybrané služby se objevovalo přibližně 40% testovaných počítačů, které neměly svůj referenční obraz. To může být způsobeno použitím pouze hodinových intervalů, ze kterých nebyla odchycena komunikace všech počítačů. Z těchto testovaných počítačů bylo 29% až 82% počítačů úspěšně označeno jako neidentifikovatelné pro různé služby a při hodnotě 0,9 prahového parametru. U počítačů vyskytujících se v obou vektorových modelech se úspěšně pozitivní identifikace pohybovala od 5% do 48% pro různé použité služby. Ze všech testovaných počítačů bylo 16% až 58% počítačů úspěšně identifikováno nebo úspěšně označeno jako neidentifikovatelné.

Kapitola 8

Závěr

Cílem této diplomové práce je seznámení s problematikou identifikace počítačů pomocí vzorů v síťovém provozu. Na základě těchto poznatku navrhnout a implementovat algoritmus pro identifikaci počítačů.

V úvodu této diplomové práce je vysvětlena základní terminologie z oblasti ochrany osobních údajů. Jsou popsány bezpečnostní funkce a hrozby pro soukromí uživatelů. Dále jsem se seznámil s možnostmi identifikace počítačů podle vzorů v síťovém provozu. Pro identifikaci počítačů se výhradně používají metody pro dolování dat. Proto jsem se zaměřil na klasifikační algoritmy z oblasti dolování dat, které již byly zdárně použity k identifikaci počítačů. Také jsem nastudoval atributy síťového provozu, které lze k identifikaci počítačů využít. Současně jsem nastudoval možné metody transformace atributů z oblasti kategorizace textových dokumentů, které umožňují zpřesnit výslednou klasifikaci.

V další části jsem navrhl nástroj pro identifikaci počítačů na základě dolovacího algoritmu Multinomial Naive Bayes. Také jsem vybral atributy síťového provozu, podle kterých je provedena identifikace počítačů. Na základě tohoto výběru jsem jako vstupní data vybral Netflow záznamy síťového provozu. V rámci návrhu jsou také diskutovány krajní případy, které mohou nastat při identifikaci počítačů pomocí navrženého nástroje.

Praktickým výstupem této práce je implementace nástroje pro identifikaci počítačů. Tento nástroj provádí identifikaci ve dvou fázích. V první fázi jsou zpracovány trénovací a testovací data, ze kterých jsou vytvořeny modely chování. V druhé fázi probíhá přiřazování stejných počítačů.

V poslední části jsou provedeny experimenty s vytvořeným nástrojem. Pro experimenty byly použity vlastní data, získána generováním síťového provozu v prostředí *Mininet* a Netflow záznamy ze sítě VUT pro ověření úspěšnosti identifikace počítačů s implementovaným nástrojem v reálném prostředí. V rámci experimentů s Netflow záznamy ze sítě VUT byla měřena úspěšnost identifikace počítačů pro různé síťové služby. Úspěšnost identifikace se pohybovala mezi 5% až 48%. U těchto záznamu bylo velké mnonžství počítačů, které se vyskytovaly pouze v testovacích datech, ale ne v trénovacích. Pro detekci počítačů, které není možné identifikovat jsem navrh prahovou funkci. U prahové funkce mohlo nastat, že úspěšně identifikovaný počítač označila jako neidentifikovatelný. Při nastavení prahové funkce tak, aby úspěšně identifikované počítače označnila jako neidentifikovatelné co nejméně, byla úspěšnost detekce neidentifikovatelných počítačů od 29% až 82%.

Na práci je možné navázat několika způsoby. První možnosti je vyběr jiné kombinace atributů síťové komunikace. Další možnosti je návrh prahovací funkce, která by lépe detekovala neidentifikovatelné počítače. Také by bylo možné se více zaměřit na předzpracování vstupních dat a stanovit kritéria pro určení, zda je o počítači získáno dostatečné množství

informací, aby ho bylo možné odlišit od ostatních.

Literatura

[1] Caligare: What is Netflow? [online], [cit. 2014-05-05].
URL http://www.caligare.com/netflow/netflow.php

[2] ISO/IEC Common Criteria for Information Technology Security Evaluation (Part 2: Security functional requirements), Verze 3.1, Revize 4. [online], Vytvořeno v září 2012 [cit. 2013-12-20].
URL
http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R4.pdf

[3] nfdump. [online], [cit. 2014-04-12].
URL http://nfdump.sourceforge.net/

[4] Banse, C.; Herrmann, D.; Federrath, H.: Tracking Users on the Internet with Behavioral Patterns: Evaluation of Its Practical Feasibility. In *Information Security and Privacy Research*, Springer, 2012, s. 235–248.

[5] Borges, J.; Levene, M.: Data mining of user navigation patterns. In *Web usage analysis and user profiling*, Springer, 2000, s. 92–112.

[6] Campbell, C.; Cristianini, N.; Shawe-Taylor, J.: Dynamically adapting kernels in support vector machines. *Advances in neural information processing systems*, ročník 11, 1999: s. 204–210.

[7] Cavoukian, A.: *Data Mining: Staking a Claim on Your Privacy*. Information and Privacy Commissioner's Report, Canada, 1998.

[8] Claise, B.: RFC 3954 Cisco Systems NetFlow Services Export Version 9. [online], Vytvořeno v říjnu 2004 [cit. 2014-04-12].
URL http://www.ietf.org/rfc/rfc3954.txt

[9] Dyer, K. P.; Coull, S. E.; Ristenpart, T.: Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *Security and Privacy (SP), 2012 IEEE Symposium on*, IEEE, 2012, s. 332–346.

[10] Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.: From data mining to knowledge discovery in databases. *AI magazine*, ročník 17, č. 3, 1996: str. 37.

[11] Herrmann, D.; Gerber, C.; Banse, C.: Analyzing characteristic host access patterns for re-identification of web user sessions. In *Information Security Technology for Applications*, Springer, 2012, s. 136–154.

[12] Herrmann, D.; Wendolsky, R.; Federrath, H.: Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, ACM, 2009, s. 31–42.

[13] Hsu, W.; Chang, C.; Lin, J.: A Practical Guide to Support Vector Classification. *Department of Computer Science National, Taiwan University, Taipei*, 2010.

[14] Hsu, W.; Lin, J.: A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, ročník 13, č. 2, 2002: s. 415–425.

[15] Kumpošt, M.: *Context information and user profiling*. Disertační práce, Masarykova univerzita, Brno, 2009.

[16] Liberatore, M.; Levine, B. N.: Inferring the source of encrypted HTTP connections. In *Proceedings of the 13th ACM conference on Computer and communications security*, ACM, 2006, s. 255–263.

[17] Manning, C. D.; Raghavan, P.; Schutze, H.: *Introduction to information retrieval*, ročník 1. Cambridge University Press Cambridge, 2008.

[18] Panchenko, A.; Niessen, L.; Zinnen, A.: Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, ACM, 2011, s. 103–114.

[19] Pfitzmann, A.; Hansen, M.: A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. Verze 0.34, srpen, 2010.
URL https://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf

[20] Salton, G.; Wong, A.; Yang, C.-S.: A vector space model for automatic indexing. *Communications of the ACM*, ročník 18, č. 11, 1975: s. 613–620.

[21] Solove, D. J.: A taxonomy of privacy. *University of Pennsylvania Law Review*, ročník 154, 2006: s. 477–564.

[22] Tavani, H. T.: Informational privacy, data mining, and the internet. *Ethics and Information Technology*, ročník 1, č. 2, 1999: s. 137–145.

[23] Vapnik, V. N.: An overview of statistical learning theory. *Neural Networks, IEEE Transactions on*, ročník 10, č. 5, 1999: s. 988–999.

[24] Weston, J.; Watkins, C.: Multi-class support vector machines. Technická zpráva, Citeseer, 1998.

[25] Witten, I. H.; Frank, E.; Hall, M. A.: *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.

Příloha A

Obsah CD

Obsah přiloženého média je následující:

- Technická zpráva ve formátu PDF a její zdrojový tvar.
- Zdrojový kód programu.
- Návod pro instalaci a spuštění programu.
- Vybrané záznamy Netflow ze simulovaného prosředí.