VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

METRIKY PRO DETEKCI ÚTOKŮ V SÍŤOVÉM PROVOZU

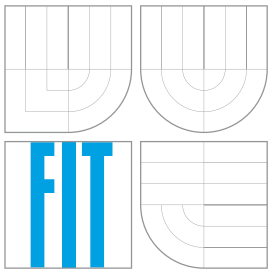
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. IVAN HOMOLIAK

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

METRIKY PRO DETEKCI ÚTOKŮ V SÍŤOVÉM PROVOZU

METRICS FOR INTRUSION DETECTION IN NETWORK TRAFFIC

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. IVAN HOMOLIAK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL DROZD

BRNO 2012

Abstrakt

Práce se zabývá návrhem a aplikací nových metrik pro detekci útoků v síťovém provozu na základě analýz již existujících metrik, analýz síťového provozu a chování známých útoků. Hlavním cílem práce je pokusit se navrhnout a implementovat takové metriky, kterými bude možné detekovat i zero day útoky.

Abstract

Publication aims to propose and apply new metrics for intrusion detection in network traffic according to analysis of existing metrics, analysis of network traffic and behavioral characteristics of known attacks. The main goal of the thesis is to propose and implement new collection of metrics which will be capable to detect zero day attacks.

Klíčová slova

honeypot, IPS, IDS, KDD Cup 99, škodlivý software, zero day útoky, buffer overflow útoky, dolovaní dat

Keywords

honeypot, IPS, IDS, KDD Cup 99, malware, zero day attacks, buffer overflow attacks, data mining

Citace

Ivan Homoliak: Metriky pro detekci útoků v síťovém
provozu, diplomová práce, Brno, FIT VUT v Brně, 2012

Metriky pro detekci útoků v síťovém provozu

Prohlášení

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

.....
Bc. Ivan Homoliak
21. mája 2012

Poděkování

Chcem poďakovať Ing. Michalovi Drozdovi za vedenie tejto práce, odbornú pomoc a konzultácie s ňou spojené. Za konzultácie a odbornú pomoc tiež ďakujem konzultantovi tejto práce Ing. Marošovi Barabasovi.

Obsah

Kapitola 1

Úvod

V súčasnej dobe je otázka bezpečnosti sieťovej prevádzky veľmi horúcou témou. Čoraz častejšie dochádza k úspešným útokom do kvalitne zabezpečených korporátnych alebo súkromných počítačových sietí a staníc. Tieto útoky spôsobujú nemalé finančné a ekonomické škody. Neschopnosť alebo nemožnosť detekcie týchto útokov existuje z dôvodu, že aktuálne bezpečnostné produkty sú založené na signatúrach známych typov útokov. V súčasnosti existujú rôzne zatemňovacie techniky, ktoré sú využívané útočníkmi na obídenie detekcie pomocou rôznych substitúcií, preskladávania kódu, komprimovanie kódu, vkladania prebytočných, nič nevykonávajúcich inštrukcií alebo transformácie celého kódu *malwarom*[1] samotným [16]. Tieto techniky sú problematické pre detekčné nástroje, ktoré sú založené na signatúrach, pretože škodlivý kód vyzerá vždy inak. Ďalšia vec ktorú treba brať v úvahu je, že samotný užitočný obsah prenášaných dát môže byť zašifrovaný a teda v rámci korporátneho systému na prevenciu vniknutia (ďalej IPS[2]) alebo systému na detekciu vniknutia (ďalej IDS[3]) bude nedešifrovateľný. Preto je potrebné zamerať sa na analýzu sieťovej komunikácie bez analýzy prenášaných užitočných dát.

1.1 Ciele práce

Hlavnou náplňou tejto práce je na základe analýz existujúcich princípov detekcií sieťových útokov navrhnúť novú kolekciu metrík, ktorými by bolo možné detekovať aj útoky *zero-day*.

V kontexte tejto práce sa pojmom metrika bude rozumieť nominálne, ordinárne alebo vektorovo vyjadrejná vlastnosť danej komunikácie, ktorá môže byť priamo alebo nepriamo odvodená zo zachytených dát tejto komunikácie. *Zero-day* útoky sú popísané ako útoky na zraniteľnosti, ktoré neboli ešte opravené. Iná definícia ich definuje ako útoky využívajúce zraniteľnosti ešte neaktualizovaných systémov v nultý deň od vydania verejnej aktualizácie (záplaty) [10].

Pri analýze a návrhu jednotlivých metrík musí byť brané v úvahu cieľové určenie tejto detekcie, ktoré má byť situované do korporátnej siete ako zariadenie analyzujúce sieťovú prevádzku a generujúce signatúry vzniknutých útokov. Na analýzu existujúcich metrík bude využitá verejne dostupná dátová kolekcia KDD Cup 1999. Ďalším cieľom práce je vytvorenie nástroja, ktorý bude extrahovať navrhnuté metriky z poskytnutých dát. Zároveň je cieľom overiť vlastnosti navrhnutej kolekcie na predanom vzorku dát z projektu MPO

[1] Softwarový program zameraný na vykonávanie škodlivej činnosti [9].
[2] Intrusion prevention system.
[3] Intrusion detection system.

AIPS. Dosiahnuté výsledky majú byť porovnané s výsledkami dosiahnutými nad dátovou kolekciou KDD Cup 1999 s využitím dolovacieho nástroja RapidMiner[4].

1.2 Štruktúra práce

V prvej časti práce sa budem venovať princípom detekcie sieťových útokov a to hlavne pomocou *honeypotov*[5]. Bude tu tiež analyzovaný princíp reprezentatívneho, v rámci práce nepriamo využitého, honeypotu Argos. V kapitole 3 bude analyzovaná dátová kolekcia KDD Cup 99 a budú diskutované jej nedostatky. Na základe týchto analýz bude navrhnutá nová dátová množina metrík, ktorá bude popísaná v kapitole 4. Implementácia extrakcie metrík bude popísaná v kapitole 5. V tejto kapitole bude tiež popísaný beh nástroja na extrakciu metrík a spôsob generovania výstupu. Potom v kapitole 6 bude uskutočnená detailná analýza navrhnutých metrík, na základe ktorej sa bude postupovať v návrhu klasifikačného modelu. Bude sa tu tiež experimentovať s rôznymi klasifikačnými metódami a procesom predspracovania vstupných dát. Súčasťou kapitoly 6 bude aj analýza dátovej kolekcie KDD Cup 1999 pre účely porovnania s výsledkami dosiahnutými nad kolekciou obsahujúcou navrhnuté metriky. V ďalšej kapitole 7 bude popísaný koncept návrhu a topológia siete, v ktorej by mohli byť aplikované výsledky tejto práce. V tejto kapitole bude tiež diskutovaný vplyv laboratórnych podmienok na efektivitu jednotlivých metrík a bude tu diskutované použitie v reálnom prostredí. Ďalej v kapitole 8 budú vyjadrené myšlienky súvisiace z návrhom ďalších metrík a ďalším rozvojom navrhnutého systému.

[4] Voľne dostupný dolovací nástroj [7].

[5] Počítačový systém explicitne slúžiaci na lákanie a zaznamenanie útočníkov, ktorí sa do neho pokúšajú preniknúť.

Kapitola 2

Princípy detekcie sieťových útokov

Podľa [15] existujú 2 typy detekcie sieťových útokov a to detekcia na základe anomálií a detekcia na základe signatúr v sieťovej prevádzke. Ďalším, modernejším spôsobom detekcie sieťových útokov sú honeypoty. Ich rozdelenie a princíp činnosti bude v nasledujúcich sekciách rozvedený. Nasledujúce informácie týkajúce sa detekcie na základe signatúr a anomálií sú čerpané z [15].

2.1 Detekcie na základe anomálií

Pri detekcii anomálií sa nesledujú známe charakteristiky útokov, ktoré by mohli byť signálmi ich výskytu ale sledujú sa abnormality v sieťovej prevádzke. Uplatňuje sa pri tom prístup, že všetko čo je abnormálne, je podozrivé. Na konštrukciu takého detektoru je potrebné sformovať úsudok, čo je normálne správanie pozorovaného subjektu a čo abnormálne.

2.1.1 Samoučiace sa systémy

Ich úlohou je sa učiť z dlhodobého pozorovania sieťovej prevádzky a budovania modelu tejto prevádzky a procesov v nej prebiehajúcich. Delia sa na berúce v úvahu časovú opakovateľnosť, a neberúce v úvahu túto opakovateľnosť. Časovo opakovateľné využívajú techniky ako Markovské rozhodovacie procesy [22] alebo *neurónové siete* [28], ktoré dokážu dôveryhodnejšie reprezentovať reálnu komunikáciu.

Časovo neopakovateľné systémy využívajú stochastický model. Môžu pracovať buď spôsobom, že sledujú normálnu sieťovú prevádzku a vytvárajú pravidlá modelu normálnej komunikácie alebo v móde detekcie, kde sledujú dodržiavanie týchto pravidiel. Ak nastane slabá zhoda, vyhodnocovaná váhovaním, tak alarmujú systém. Iný spôsob akým môžu pracovať je zbieranie popisných štatistík sieťovej prevádzky do profilu a konštruujú z nich vektor vzdialeností pre skúmanú prevádzku a skúmaný profil. Ak je vzdialenosť dostatočne veľká, tak alarmujú systém.

2.1.2 Programované systémy

V tejto variante je potrebný programátor, ktorý naučí systém detekovať určité udalosti, ktoré sú anomáliami. Tým pádom užívateľ programátor vkladá do systému predsudok, čo je považované za abnormálne a má spôsobiť porušenie bezpečnosti. Tieto systémy sa delia na vychádzajúce z deskriptívnych štatistík a implicitne zakazovacie. Tie, ktoré vychádzajú z deskriptívnych štatistík, zbierajú parametre a informácie o systéme ako sú napríklad počet

neúspešných prihlásení, počet sieťových pripojení, počet príkazov s chybovým návratovým kódom. Hlavnou myšlienkou implicitne zakazovacích systémov je uviesť stavové okolnosti systému do neškodného módu a označiť všetky odchýlky od tohto módu za útoky. Je tu možné vidieť korešpondenciu s implicitne zakazovacou bezpečnostnou politikou.

2.2 Detekcie na základe signatúr

V oblasti systémov založených na detekcii signatúr, je rozhodnutie vytvorené na základe modelu škodlivého procesu a stôp, ktoré môže zanechať útok na kompromitovanom systéme. Malo by byť poznamenané, že tieto detektory sa snažia detekovať prítomnosť škodlivej aktivity bez ohľadu na normálnu komunikáciu prebiehajúcu v pozadí. Práve tento fakt kladie striktné požiadavky na model detekcie preniknutia [15], v ktorom musia byť veľmi presne nadefinované všetky detaily signatúry. Všetky systémy z tejto kategórie sú explicitne programované a nedochádza tu k učeniu počas behu systému. Delia sa na stavovo modelované, expertné, založené na pravidlách a založené na reťazcovom porovnávaní.

2.2.1 Stavovo modelované systémy

Tieto systémy reprezentujú útok ako množinu stavov, z ktorých každý musí nastať na to aby bolo možné prehlásiť, že došlo k útoku. K týmto stavom existujú modely časových rád. Existujú dve podtriedy z tejto kategórie a to stavovo prechodové a založené na Petriho sieťach [23]. V stavovo prechodových musia stavy pri výskyte útoku tvoriť postupnosť, ktorá musí byť prejdená od začiatku po koniec. V druhej podtriede Petriho sietí môže mať graf prechodu jednotlivými stavmi obecnejšiu štruktúru.

2.2.2 Expertné systémy

Expertný systém využíva preddefinované pravidlá, ktoré sú signatúrami útokov. Veľmi často sa jedná o sekvencie pravidiel, ktoré musia byť splnené na to, aby bol detekovaný útok. Problémom týchto systémov je dlhá doba vykonávania analýz vzhľadom na časté použitie výkonných mechanizmov.

2.2.3 Reťazcové porovnávanie

Systémy založené na reťazcovom porovnávaní pracujú s veľmi jednoduchým porovnávaním podreťazcov v texte prenášanom medzi systémami. Táto metóda nie je flexibilná ale je jednoduchá na pochopenie a môže využiť potenciál existencie mnohých veľmi efektívnych algoritmov na prehľadávanie textových reťazcov.

2.2.4 Systémy založené na pravidlách

Tieto systémy sú podobné niektorým jednoduchým expertným systémom. Sú v porovnaní s nimi rýchlejšie. Pravidlá, na ktorých sú založené, majú deklaratívny charakter. Mnohokrát je potrebné optimalizovať jednotlivé pravidlá ako napríklad zložitý výraz sa transformuje na stromovú reprezentáciu a pri vyhodnocovaní dochádza k eliminácii vyhodnocovania niektorých podvýrazov [20].

2.3 Detekcia útokov honeypotmi

Množstvo informácií, ktoré sú schopné získavať zo sieťovej prevádzky IDS a IPS systémy je čoraz väčšie. Taktiež sa zväčšuje aj počet protokolov aplikačnej vrstvy modelu TCP/IP[1] (CITE RFC TCP IP) zavádzajúcich šifrovanie z dôvodu, aby nebolo možné odpočúvať užitočné dáta sieťovej komunikácie. IDS a IPS systémy tiež trpia vysokou mierou *false positive*[2], ktoré znižujú potenciál ich použitia [15].

Honeypot je podrobne monitorovaný zdroj, ktorý má byť čo najviac skúmaný, atakovaný a kompromitovaný. Inak povedané, honeypot je zdroj informačného systému, ktorého hodnota spočíva v neautorizovanom či nelegálnom využití jeho samotného ako zdroja [13]. Hodnota honeypotu je určená na základe množstva informácií, ktoré z neho môžu byť získané. Sledovanie dát, ktoré vstupujú a vystupujú z honeypotu nám umožňuje získať informácie, ktoré nie sú dostupné pre IDS alebo IPS. Napríklad môžu byť sledované stlačené klávesy počas interaktívneho sedenia aj keď je zapnuté šifrovanie na ochranu sieťovej prevádzky. Na detekciu škodlivého správania IDS alebo IPS potrebujú signatúry známych útokov a často zlyhávajú pri detekcii existujúcich útokov, ktoré zmenili svoje správanie a tým aj signatúry. Ďalšou výhodou honeypotov je, že sú schopné detekovať zraniteľnosti, ktoré ešte neboli objavené.

Keďže honeypot nemá žiadnu produkčnú hodnotu, akýkoľvek pokus o nadviazanie spojenia naň je považovaný za podozrivý, čo vyplýva nepriamo z definície. Forézna analýza dát nazbieraných honeypotmi je menej náchylná na false positive na rozdiel od IDS a IPS, pretože väčšina dát nazbieraných týmito systémami sú priamo nazbierané počas reálnych útokov. Ďalšou výhodou honeypotov je, že môžu byť spustené v koexistencii s akýmkoľvek operačným systémom a s akýmikoľvek spustenými službami.

Vysokointeraktívny honeypot poskytuje reálny systém, s ktorým môže prichádzať do styku útočník. V kontraste s ním stojí nízkointeraktívny honeypot, ktorý simuluje len niektoré časti (napríklad sieťový model). Vysokointeraktívny honeypot môže byť kompromitovaný – môže útočníkovi umožniť získať plný prístup k systému a využiť systém honeypotu na ďalšie sieťové útoky. Nízkointeraktívne honeypoty môžu simulovať len také služby, ktoré nemôžu byť využité na získanie plného prístupu k systému. Nízkointeraktívne honeypoty sú viac obmedzené, no ich výhodou môže byť napríklad získanie informácií na vyššej vrstve (napr. naučenie sa o sieťových skenovaniach alebo aktivite červov). Môžu byť tiež použité na analýzu chovania spamerov alebo aktívne získavanie opatrení proti červom. Ani jeden z týchto prístupov nie je nadradený druhému. Každý má svoje výhody a nevýhody.

Je potrebné rozlišovať medzi fyzickými a virtuálnymi honeypotmi. Fyzický honeypot je skutočný stroj na sieti, ktorý má svoju vlastnú IP adresu. Virtuálny honeypot je simulovaný pomocou iného stroja.

Pri získavaní informácií o sieťových útokoch a skenovaniach siete je dôležitým faktorom počet nasadených honeypotov, ktorý ovplyvňuje kvalitu a kvantitu nazbieraných dát. Vhodným príkladom je monitorovanie aktivity červov založených na HTTP[3]. Tieto červy môžu byť identifikované len v prípade, ak dôjde ku kompletnému TCP handshaku a poslaniu užitočných dát útoku. Preto honeypot, ktorý neemuluje zraniteľnú webovú službu, nemôže ani tieto informácie zhromažďovať. Potom čím viacej honeypotov je nasadených, tým väčšia je pravdepodobnosť, že jeden z nich bude kontaktovaný červom [25]. Nasledujúce informácie týkajúce sa rozdelenia honeypotov a popisu Argosu budú čerpané z [25].

[1] Vrstvový model architektúry sieťovej komunikácie.
[2] Falošné poplachy (detekcia anomálií alebo signatúr aj keď sa nejednalo o útok).
[3] Hypertext transfer protocol. http://tools.ietf.org/html/rfc2616.

2.3.1 Vysokointeraktívne honeypoty

Vysokointeraktívny honeypot je konvenčný počítačový systém, napríklad počítač smerovač alebo prepínač. Tento systém nemá žiadne konvenčné použitie alebo úlohu v rámci siete. Taktiež nemá regulárnych aktívnych užívateľov. Preto nemá žiadne nezvyčajné procesy a negeneruje žiadnu sieťovú prevádzku (okrem skutočných služieb a prevádzkou s nimi spojenou). Z týchto predpokladov sa vychádza pri detekcii útokov. Každá interakcia s honeypotom je podozrivá a potenciálne môže byť škodlivou akciou, a preto je všetka sieťová prevádzka smerujúca do honeypotu a smerujúca von z honeypotu logovaná do súborov pre neskoršie analýzy.

Je možné kombinovať niekoľko honeypotov do siete honeypotov nazvanej honeynet. Obvykle honeynet pozostáva z niekoľkých honeypotov rôzneho typu - rôzne platformy a operačné systémy. Toto členenie umožňuje nezávislé a paralelné zbieranie dát o rozličných typoch útokov, a preto je zvyčajne možné študovať informácie o útokoch do hĺbky a týmto získať kvalitatívne výsledky o správaní útočníkov. Jedným z kľúčových elementov honeynetu je honeywall – čo je sieťové zariadenie fungujúce na linkovej vrstve modelu TCP/IP, ktoré premosťuje honeywall od zvyšku siete. Toto zariadenie zmenšuje ohrozenie ostatných staníc v sieti a umožňuje zachytávať dáta pre analýzu. Každá prichádzajúca a odchádzajúca prevádzka musí prejsť honeywallom. Informácie sú zachytávané využitím rôznych metód ako sú pasívne monitorovacie nástroje, IDS notifikácie, logy firewallu[4] atď. Aktivity útočníka sú riadené na úrovni sieťovej vrstvy modelu TCP/IP a všetky odchádzajúce spojenia filtrované pomocou IPS a obmedzovačmi spojení.

Jedným z problémov vysokointeraktívnych honeypotov sú vyššie nároky na údržbu. Honeypot by mal byť dôkladne monitorovaný a malo by byť detailne skúmané, čo sa v ňom deje. Kompletná analýza incidentov môže zabrať hodiny ba dokonca celé dni po okamžik, kedy sa podarí plne pochopiť, čo útočník zamýšľal dosiahnuť. Vysokointeraktívne honeypoty môžu byť plne kompromitované. Sú na nich spustené reálne operačné systémy. Útočník môže interagovať s reálnymi službami a reálnym systémom, čo umožňuje zachytiť rozsiahle informácie súvisiace s hrozbami, ktoré predstavuje. Je možné zachytiť exploity[5] útočníkov pri získaní neautorizovaného prístupu, monitorovať stlačené klávesy, odhaliť ich použité nástroje a odhaliť ich motívy. Nevýhodou vysokointeraktívnych honeypotov je, že zvyšujú risk, pretože útočník môže potenciálne ovládnuť celý operačný systém a prostredníctvom neho predstavovať potenciálne nebezpečenstvo pre produkčné systémy v sieti. Hlavnou nevýhodou vysokointeraktívnych honeypotov je ich náročnosť na výpočtové zdroje a problémy s ich škálovaním a nasadením v reálnej sieti.

2.3.2 Nízkointeraktívne honeypoty

V kontraste s predchádzajúcou kategóriou, nízkointeraktívne honeypoty emulujú služby, sieťový model alebo iné aspekty reálneho systému. Umožňujú útočníkovi obmedzenú interakciu s cieľovým systémom a umožňujú hlavne získanie kvantitatívnych informácií o útokoch. Napríklad emulovaný HTTP server môže implementovať len podmnožinu špecifikácie protokolu HTTP. Interakcia s útočníkom je implementovaná na takej úrovni, aby postačila na zmätenie útočníka alebo automatizovaného nástroja, ktorý napríklad hľadá špecifický súbor potrebný na kompromitovanie systému. Výhodou nízkointeraktívnych honeypotov je ich jednoduchá údržba. Pre uvedenie do prevádzky nízkointeraktívneho honeypotu je

[4] Filter sieťovej komunikácie na úrovni sieťovej a transportnej vrstvy modelu ISO/OSI.

[5] Kód využívajúci zraniteľnosť služby pre škodlivé ciele.

postačujúce jeho spustenie. K zbieraniu dát dochádza automaticky. Tieto dáta môžu byť informácie o propagovaní sieťových červov alebo skenovania spamerov na otvorené možnosti prenosu. Pre tento druh honeypotu je inštalácia všeobecne jednoduchšia: stačí nainštalovať a nakonfigurovať nástroj. Na rozdiel od vysokointeraktívnych honeypotov kde žiadané prispôsobenie na dané prostredie.

Nízkointeraktívne honeypoty môžu byť primárne použité na zbieranie štatistických dát a na zhromaždenie vysokoúrovňových informácií o vzoroch použitých v útokoch (signatúrach). Je možné ich použiť aj ako IDS, keďže poskytujú skoré varovania na výskyt útokov. Dokonca môžu byť využité na odlákanie útočníkov od produkčných strojov korporátnej siete. Ich ďalšie použitie je na detekciu červov a naučenie sa o prebiehajúcich útokoch. Podobne ako u vysokointeraktívnych honeypotov aj tu môžu byť honeypoty použité na vytvorenie siete honeypotov zvanej honeynet.

Útočník nie je schopný získať plnú kontrolu nad systémom, keďže neinteraguje s reálnou službou. Nízkointeraktívne honeypoty poskytujú plne ovládateľné. kontrolovateľné a monitorovateľné prostredie. Preto akékoľvek riziko spojené s možnosťou preniknúť do systému, pod ktorým honeypot beží, je minimálne.

2.3.3 Fyzické honeypoty

Ďalším možným rozdelením v oblasti honeypotov rozlišujeme medzi fyzickými a virtuálnymi honeypotmi. Fyzický honeypot znamená, že beží na fyzickom stroji. Bežiaci na fyzickom stroji mnohokrát implikuje vysokú interakciu a tým umožní kompletnú kompromitovanosť systému. Táto kategória honeypotov je typicky náročná na inštaláciu a údržbu. Nasadenie veľkého počtu honeypotov v korporátnej sieti je náročné na fyzické zdroje a preto je výhodnejšie zaviesť virtuálne honeypoty.

2.3.4 Virtuálne honeypoty

Hlavnou výhodou používania tohto typu honeypotov je škálovateľnosť a jednoduchosť údržby. Môže existovať niekoľko honeypotov na jednom fyzickom stroji. V porovnaní s fyzickými honeypotmi je tento prístup odľahčený. Namiesto nasadenia fyzického stroja, ktorý sa bude správať ako honeypot, môže byť nasadených niekoľko virtuálnych strojov, ktoré reprezentujú jednotlivé honeypoty. Toto vedie na jednoduchú udržovateľnosť a nižšie fyzické nároky. Obyčajne môžu byť využité virtualizačné nástroje ako VMware[6], USL[7], Virtualbox[8], QEMU[9]. Hlavný aspekt, ktorý je potrebný brať do úvahu je, že virtuálny honeypot je simulovaný iným strojom, ktorý zodpovedá za preposielanie sieťovej prevádzky určenej honeypotu a prevádzky, ktorú vytvára honeypot.

2.3.5 Argos

Jedná sa o nový nástroj, ktorý spadá do kategórie virtuálnych vysokointeraktívnych shadow honeypotov [14] a bol vyvinutý výskumníkmi z Vrije Universiteit Amsterdam v Holandsku. Nástroj zvaný Argos, je schopný detekcie *zero-day* útokov. Využíva techniku zvanú *dynamic taint analysis* [24] na monitorovanie honeypotu. Ako prvý krok, všetky dáta prijaté cez sieť sú označené ako *tainted*. Použitie týchto označených dát je potom sledované aj v rámci

[6] Virtualizačný nástroj. http://www.vmware.com/cz/.
[7] Virtualizačný nástroj pre Linux. http://user-mode-linux.sourceforge.net/.
[8] Virtualizačný nástroj. https://www.virtualbox.org/.
[9] Strojový emulátor a virtualizačný nástroj. http://wiki.qemu.org/Main_Page.

operačnej pamäte honeypotu. Akonáhle je detekované ovplyvnenie toku vykonávajúceho sa programu (napr. pomocou inštrukcie JUMP), Argos to detekuje a vygeneruje odtlačok pamäte reprezentujúci aktuálny stav pamäte pri útoku.

V porovnaní s inými virtuálnymi honeypotmi, Argos zastupuje mierne odlišný prístup. Namiesto čistého vykonávania virtuálneho stroja ho aj dôkladne monitoruje a snaží sa detekovať bod v čase, kedy došlo k úspešnému kompromitovaniu virtuálneho stroja.

Argos je modul QEMU, čo je emulátor operačných systémov, ktorý na emuláciu využíva techniku dynamického prekladu. Výkonnosť je však nižšia v porovnaní s ostatnými virtualizačnými nástrojmi. Argos pridáva emulátoru QEMU možnosť dynamickej analýzy *tainted* dát.

2.3.5.1 Dynamická analýza *tainted* dát

Dynamická analýza *tainted* dát je jadrom Argosu. Táto technika je založená na pozorovaní toku programu a jeho ovplyvnení útočníkom. Toto môže byť dosiahnuté podtečením alebo pretečením zásobníku programu. Útočník pošle špeciálne formovaný textový reťazec, ktorý prepíše citlivé pamäťové miesta a tieto dáta ovplyvnia tok programu napr. skokom na pamäťovú oblasť, kde sa nachádza útočníkom poskytnutý program. V tomto okamžiku prichádza na radu dynamická analýza *tainted* dát. Všetok externý vstup programu je označený ako *tainted* a počas analýzy je používanie *tainted* premenných sledované a kontrolované.

Kapitola 3

Analýza KDD Cup 99

Dátová kolekcia pochádzajúca z roku 1999 je zatiaľ jedinou verejne dostupnou kolekciou dát súvisiacimi so sieťovými útokmi. Dáta kolekcie KDD Cup 99 boli získané simulovanými útokmi na high-end honeypoty v armádnom prostredí. Jej pôvodné použitie bolo pre súťaž v dolovaní dát, čo naznačuje aj názov. V KDD 99 dátovej množine sú jednotlivé vzorky klasifikované do 24 tried, kde každá trieda reprezentuje iný typ útoku [30].

3.1 Popis kolekcie

Táto dátová množina je pripravená v rámci práce [27] a je postavená na dátach zachytených v rámci DARPA'98 IDS programu [26]. DARPA'98 obsahuje 4 GB komprimovaných binárnych TCPdump[1] dát zachytených za 7 týždňov nepretržitej sieťovej prevádzky (trénovacie dáta). Tieto dáta môžu byť separované do 5 miliónov záznamov reprezentujúcich jednotlivé spojenia, kde každé spojenie obsahuje dáta o veľkosti 100 B. Pre zostavenie testovacích záznamov boli použité iné dáta získané počas dvojtýždňovej nepretržitej sieťovej prevádzky. Počet záznamov testovacej množiny je vyše 2 milióny. Každý vektor trénovacej a testovacej množiny obsahuje 41 prvkov, ktoré majú definičný obor buď ordinárny z množiny reálnych čísel alebo nominálny kategorický. Každý záznam obsahuje návestie buď normálnej komunikácie alebo útoku, kde útoky sú delené do 24 tried v trénovacej množine a do 24 plus ďalších 14 tried v testovacej množine.

Simulované útoky spadajú do jednej zo 4 nasledujúcich kategórií [29]:

1. **Denial of service (DoS)** - útok, kde útočník zahltí výpočtový alebo pamäťový zdroj kompromitovaného stroja natoľko, že tento stroj nie je schopný obsluhovať legitímne požiadavky alebo nie je schopný umožniť legitímnym užívateľom prihlásenie či prístup k tomuto stroju.

2. **User to root (U2R)** - je trieda zneužitia, kde útočník začína s normálnym užívateľským účtom na kompromitovanom systéme (získaným buď odpočúvaním hesiel, slovníkovými útokmi alebo sociálnym inžinierstvom) a je schopný zneužiť určitú zraniteľnosť na získanie administrátorského účtu v systéme.

3. **Remote to local (R2L)** - nastáva keď útočník, ktorý má možnosť komunikovať s kompromitovaným strojom cez sieť ale nemá na ňom prístup ku žiadnemu účtu, zneužije zraniteľnosť na získanie prístupu k lokálnemu užívateľovi toho stroja.

[1] Paketový analyzátor pracujúci v príkazovom riadku. http://www.tcpdump.org/.

4. **Probing** - je pokus o získanie informácií o počítačovej sieti pre účely analýzy zraniteľností nachádzajúcich sa na jednotlivých strojoch.

Je dôležité spomenúť, že testovacie dáta sa neriadia rovnakým rozložením pravdepodobnosti ako trénovacie a obsahujú aj špecifické typy útokov neobsiahnutých v trénovacích dátach čo robí dolovanie viac realistickým. Niektorí odborníci na bezpečnostné útoky veria, že väčšina nových útokov sú len varianty známych útokov a signatúry známych útokov môžu byť použité na detekciu nových variant. Prvky kolekcie KDD Cup 99 môžu byť klasifikované do troch skupín [29]:

1. **Základné prvky** - táto kategória zahŕňa všetky atribúty, ktoré môžu byť priamo extrahované z TCP/IP spojenia. Jedná sa napríklad o IP adresy alebo porty skúmaného spojenia.

2. **Prevádzkové prvky** - táto kategória obsahuje prvky vypočítané s ohľadom na interval okna a sú delené do 2 skupín: **a) prvky s rovnakým cieľom** - popisujú spojenia len v intervale uplynulých 2 sekúnd, ktoré majú rovnakú cieľovú IP adresu ako súčasné spojenie a vypočítavajú štatistiky so správaním protokolu, služieb a pod. **b) prvky s rovnakou službou** - popisujú spojenia len v intervale uplynulých 2 sekúnd, ktoré majú rovnaký typ služby ako aktuálne spojenie. Tieto typy prevádzkových prvkov sú označované ako časovo založené. Problémom je, že mnoho skenovacích útokov, skenujúcich v dlhšom intervale ako 2 sekundy, a preto takéto útoky neprodukujú vzory reprezentujúce preniknutie. Na vyriešenie tohto problému sa zaviedol prístup kde sú tieto prvky prepočítané na 100 spojení. Tieto prvky sú označované ako založené na spojeniach.

3. **Obsahové prvky** - na rozdiel od mnohých DoS alebo skenovacích útokov, útoky typu R2L alebo U2R nemajú žiadne frekventované sekvenčné vzory útoku. Signatúry R2L a U2R útokov sú zabudované do dátovej časti paketov a vyskytujú sa v rámci jedného spojenia. Príkladom metriky spadajúcej do tejto kategórie je počet neúspešných prihlásení do kompromitovaného systému.

3.2 Nedostatky kolekcie

Nedostatky kolekcie KDD Cup 99 boli diskutované v [29]. Jedná sa najmä o to, že pôvodná sada obsahuje okolo päť miliónov záznamov, kde je väčšina z nich len duplikovaných ako v trénovacej sade, tak aj v testovacej. Počet replikovaných záznamov je v trénovacej sade 78% a v testovacej sade 75%. Ďalším nedostatkom je, že neobsahuje rovnomerné zastúpenie vzoriek v rámci jednotlivých tried, čo viedlo na pomerne slabé výsledky získané rôznymi dolovacími metódami využívajúcimi ako napr. neurónové siete tak aj SVM[2]. Preto v rámci práce [29] došlo k vyfiltrovaniu pôvodnej dátovej množiny o duplicitné záznamy a tiež boli opravené nekonzistencie diskutované v tejto publikácii ako napr. typografické chyby. Upravená dátová množina je dostupná na [12]. Trénovacia množina obsahuje 125973 vzoriek a testovacia množina 22544 vzoriek. Na [12] sú dáta k dispozícii zaradené ako do pôvodných 24 tried, tak aj do 2 tried, kde jedna trieda reprezentuje, že k útoku došlo a druhá trieda reprezentuje, že sa jednalo o legitímnu komunikáciu.

Ďalšia kritika diskutovaná v [29] zdôrazňuje, že žiadne analytické ani experimentálne validácie nesprávneho alarmovacieho pomeru neboli uskutočňované a samotný obsah dát,

[2]Support Vector Machines. http://www.support-vector.net/.

z ktorých bola kolekcia vytvorená sa nezdá byť podobný ako z prevádzky bežných reálnych sietí. Iným problémom môže byť, že TCPdump použitý na ukladanie sieťových dát sa môže veľmi ľahko dostať do stavu kedy kvôli preťaženosti zahadzuje prichádzajúce pakety. Žiadny výskum výskytu tejto možnosti sa však doposiaľ neuskutočnil. Nie je tiež stanovená exaktná definícia pre útok. Napríklad skenovanie nie je nevyhnutne útokom, až kým počet iterácií nepresiahne určitý prah. Taktiež paket, ktorý spôsobil pretečenie zásobníku, nemusí vždy byť príčinou útoku.

Navrhnuté modifikácie a filtrovania nediskutujú nedostatky metrík tejto kolekcie. Zaoberajú sa len úpravami súvisiacimi s dolovaním dát. Je potrebné zdôrazniť, že dáta, ktoré táto kolekcia obsahuje, vznikli simulovanými útokmi a nemusia reprezentovať reálnu situáciu.

3.3 Nedostatky metrík kolekcie

Aby boli metriky vektoru reprezentujúceho spojenie v praxi použiteľné je potrebné uvažovať, že užitočný obsah sieťovej komunikácie je zašifrovaný a preto sa môže vychádzať len z metainformácií daného spojenia. Tento fakt dátová kolekcia KDD Cup 99 nerešpektuje, keďže používa obsahové metriky, ktoré predpokladajú analýzu obsahovej časti paketov a uvažuje tiež dôsledok, ktorý na kompromitovanom systéme spôsobili. Vzhľadom na cieľovú architektúru nasadenia, ktorej sa počas tejto práce držíme, nemôžeme uvažovať s dôsledkami útokov, ktoré predpokladajú dešifrovanie obsahu paketov (analyzátor sieťovej prevádzky je zariadenie pracujúce v promiskuitnom režime a analyzuje cudziu komunikáciu, ktorej nie je účastníkom a preto ju nevie dešifrovať). Zhrnutie nedostatkov kolekcie je uvedený v nasledujúcich bodoch:

1. predpoklad analýzy obsahu paketov (tým predpoklad dešifrovania),
2. analýza dôsledku na cieľovom systéme (počet neúspešných prihlásení, úspešnosť prihlásenia, získanie prístupu k root shellu, počet vytvorených a pristupovaných súborov, počet príkazov sieťových služieb atď.),
3. kontext komunikácie uvažuje len pakety odchytené za posledné 2 sekundy (počet spojení s rovnakou IP ako je aktuálna IP za posledné 2 sekundy, počet spojení na rovnakú službu ako je aktuálna za posledné 2 sekundy),
4. delenie niektorých metrík na same-host a same-service kategórie – v same-service metrikách sú zahrnuté aj nerelevantné informácie legitímnych spojení, a preto metrika počet percent spojení, ktoré majú SYN[3] chyby je rovnako ohodnotená aj pre škodlivé a aj pre legitímne spojenia.

[3]Transmission Control Protocol. http://www.networksorcery.com/enp/protocol/tcp.htm.

Kapitola 4

Návrh nových metrík

Kolekcia metrík, ktorá bude popísaná v tejto časti práce bola navrhnutá v spolupráci s vedúcim a konzultantom tejto práce. Vzhľadom na to, že niektoré doposiaľ existujúce a používané metriky môžu byť potenciálne významné pri klasifikácii sieťových útokov, tak budú tiež do tejto kolekcie zahrnuté. Všetky metriky sa budú vzťahovať k TCP spojeniam a v mnohých prípadoch budú separované podľa smeru toku dát v rámci jedného spojenia. Zameranie na TCP spojenia bolo navrhnuté z niekoľkých dôvodov. V prípade analýzy aj UDP komunikácií by bolo problematické identifikovať, ktoré pakety patria ku ktorej komunikácii, keďže môže kedykoľvek dôjsť k zmene portu alebo IP adresy útočníka v rámci jedného útoku. Pri TCP spojeniach musí dôjsť vždy najskôr k *3-way handshake*[1]. Až po jeho uskutočnení, je TCP spojenie zahájené. Toto TCP spojenie potom prebieha na tých istých portoch a IP adresách, preto nie je problém stanoviť, ktoré pakety patria ktorému spojeniu. Trochu väčším problémom je ukončenie spojenia prostredníctvom 3-way endshaku[2] preto, lebo nie vždy k tomuto ukončeniu pri istých útokoch musí dôjsť. V mnohých útokoch typu *buffer overflow* sa aktuálne spojenie neukončuje korektne, ale ďalšia komunikácia s útočníkom pokračuje na backdoor porte[3], ktorý si útočník sám otvoril. Pri návrhu metrík je tento fakt potrebné tiež zohľadniť.

4.1 Zavedené konvencie

Pre prípad, že súčasťou útoku môže byť viacej paralelne prebiehajúcich spojení, je potrebné preskúmať a do metrík zahrnúť aj okolie daného spojenia. Týmto okolím sa myslia informácie získane z aktuálne prebiehajúcich spojení. Môže tu byť stanovený časový horizont, ktorý určí ktoré spojenia brať v úvahu a ktoré nie. Aktuálne prebiehajúce spojenia budú označované ako **kontext** analyzovaného spojenia. Pre návrh metrík bolo potrebné tiež uvažovať dáta, ktoré sú získateľné v rámci aplikácie TCPdump a uvažovať len metriky, ktoré sa dajú z týchto dát skonštruovať. V rámci jednotlivých metrík sa bude uvažovať označenie zdrojových a cieľových paketov z pohľadu strany, ktorá spojenie iniciovala.

[1] Troj cestný handshake [8].
[2] Troj cestný endshake [8].
[3] Port otvorený útočníkom na kompromitovanom stroji pre jeho zlomyselné potreby.

4.2 Navrhnutá kategorizácia

Pre účely lepšej štruktúrovanosti kolekcie metrík bola navrhnutá kategorizácia, ktorá berie v úvahu charakter a príznačné vlastnosti jednotlivých metrík. Metriky boli rozdelené do piatich kategórií a to štatistické, dynamické, lokalizačné, distribujúce a behaviorálne. V nasledujúcich sekciách budú uvedené a popísané navrhnuté metriky s ohľadom na zavedenú kategorizáciu.

4.3 Štatistické metriky

V rámci tejto kategórie metrík boli identifikované najmä štatistické vlastnosti analyzovaných spojení. Zoznam metrík s ich popisom je uvedený v tabuľkách č. 4.1 a 4.2. Predpokladom pre tieto metriky je, že sa implicitne neberie do úvahy kontext daného spojenia (paralelne prebiehajúce iné spojenia). Pri návrhu niektorých metrík bolo čerpané z metrík použitých v rámci práce [19]. Ak metrika rozlišuje prichádzajúce a odchádzajúce pakety, tak je to explicitne uvedené v označení metriky (in/out <=> down/upl <=> dst/src).

Označenie metriky	**Stručný popis metriky**	**Poznámka**
BytesPerSecUpload	Počet B za sekundu v smere von.	
BytesPerSecDownload	Počet B za sekundu v smere dnu.	
BytesPerSessUpload	Počet prenesených B v rámci TCP session v smere von.	
BytesPerSessDownload	Počet prenesených B v rámci TCP session v smere dnu.	
Bytes3WH2FIN	Počet všetkých prenesených B od začiatku komunikácie do konca.	V prípade, že spojenie nemá korektné ukončenie, tak sa berú v úvahu pakety do 2 minút od začiatku spojenia.
BytesTCPSess	Počet prenesených B od začiatku komunikácie do konca v rámci TCP spojenia.	
BytesTCPOverhead	Počet prenesených B na zostavovanie a ukončovanie TCP spojenia.	Metrika je získaná z metrík Bytes3WH2FIN a BytesTCPSess
MedPktLenSrc	Medián veľkosti zdrojového paketu.	
ModPktLenSrc	Modus veľkosti zdrojového paketu.	
MeanPktLenSrc	Priemer veľkosti zdrojového paketu.	
SigPktLenSrc	Smerodajná odchýlka veľkosti zdrojového paketu.	
SumPktLenSrc	Suma veľkosti všetkých zdrojových paketov.	
MedPktLenDst	Medián veľkosti všetkých cieľových paketov.	
ModPktLenDst	Modus veľkosti všetkých cieľových paketov.	
MeanPktLenDst	Priemer veľkosti všetkých cieľových paketov.	
SigPktLenDst	Smerodajná odchýlka veľkosti všetkých cieľových paketov.	
SumPktLenDst	Suma veľkosti všetkých cieľových paketov.	
cntPktSFrom	Počet zdrojových paketov za sekundu.	
cntPktSTo	Počet cieľových paketov za sekundu.	
sumPktFrom	Počet všetkých zdrojových paketov.	
sumPktTo	Počet všetkých cieľových paketov.	
sumSYNPerSess	Počet SYN paketov v rámci spojenia.	
sumSACKPerSess	Počet SYN + ACK paketov v rámci spojenia.	
ratInoutB	Pomer odoslaných a prijatých B.	
ratInoutPkt	Pomer odoslaných a prijatých paketov.	
medTTLIn	Medián hodnoty TTL pre zdrojové pakety.	
modTTLIn	Modus hodnoty TTL pre zdrojové pakety.	
sigTTLIn	Smerodajná odchýlka hodnoty TTL pre zdrojové pakety.	
meanTTLIn	Priemer hodnoty TTL pre zdrojové pakety.	

Tabuľka 4.1: Štatistické metriky (časť 1).

Označenie metriky	Stručný popis metriky	Poznámka
sumTTLIn	Suma hodnoty TTL pre zdrojové pakety.	
medTTLOut	Medián hodnoty TTL pre cieľové pakety.	
modTTLOut	Modus hodnoty TTL pre cieľové pakety.	
sigTTLOut	Smerodajná odchýlka hodnoty TTL pre cieľové pakety.	
meanTTLOut	Priemer hodnoty TTL pre cieľové pakety.	
sumTTLOut	Suma hodnoty TTL pre cieľové pakety.	
cntDataPktIn	Počet cieľových dátových paketov.	Dátový paket sa myslí paket s nenulovou dĺžkou obsahu.
cntNondPktIn	Počet cieľových nedátových paketov.	Nedátový paket sa myslí paket s nulovou dĺžkou obsahu.
cntDataPktOut	Počet zdrojových dátových paketov.	Dátový paket sa myslí paket s nenulovou dĺžkou obsahu.
cntNondPktOut	Počet zdrojových nedátových paketov.	Nedátový paket sa myslí paket s nulovou dĺžkou obsahu.
modTosIp	Modus hodnoty TOS v IP hlavičke paketov spojenia.	
medTosIp	Medián hodnoty TOS v IP hlavičke paketov spojenia.	
sigTCPHdrLen	Smerodajná odchýlka dĺžky hlavičky TCP paketov spojenia.	
meanTCPHdrLen	Priemerná dĺžka hlavičky TCP paketov spojenia.	
sumTCPHdrLen	Suma dĺžok hlavičiek TCP paketov spojenia.	
modTCPHdrLen	Modus dĺžky hlavičky TCP paketov spojenia.	
medTCPHdrLen	Medián dĺžky hlavičky TCP paketov spojenia.	
hasFragIp	Identifikácia výskytu fragmentácie v rámci paketov spojenia.	
sumFragPkt	Počet fragmentovaných paketov.	
sumNfragPkt	Počet nefragmentovaných paketov.	
ratFragNfrag	Pomer počtu fragmentovaných ku nefragmentovaným paketom.	

Tabuľka 4.2: Štatistické metriky (časť 2).

4.4 Dynamické metriky

U tejto kategórie metrík boli sledované najmä dynamické vlastnosti spojenia súvisiace s rýchlosťou prenosu a chybovosťou prenosového kanálu (či už simulovanou alebo skutočnou). V mnohých metrikách sa berie do úvahy kontext spojenia, čo je v poznámke označené skratkou K. Kde sa kontext spojenia do úvahy neberie, je použitá skratka NK. Navrhnuté metriky sú uvedené v tabuľke č. 4.3. Rozdiel medzi štatistickými a dynamickými metrikami je možné znázorniť na jednej inštancii TCP spojenia vykonávajúcej tú istú výmenu paketov len v inom kontexte ostatných komunikácií a s inými možnými opakovanými prenosmi alebo opakovanými nadväzovaniami či ukončovaniami TCP spojenia. Ak metrika rozlišuje prichádzajúce a odchádzajúce pakety, tak je to explicitne uvedené v označení metriky (in/out).

Označenie metriky	Stručný popis metriky	Poznámka
BPerSecIn	Počet prenesených B za sekundu v smere dnu. Reprezentácia rýchlosti.	K.
BPerSecOut	Počet prenesených B za sekundu v smere von. Reprezentácia rýchlosti.	K.
BPerSesIn	Počet prenesených B počas TCP session v smere dnu.	K.
BPerSesOut	Počet prenesených B počas TCP session v smere von.	K.
PktPerSIn	Počet paketov za sekundu v smere dnu.	K.
PktPerSOut	Počet paketov za sekundu v smere von.	K.
PktPerSesIn	Počet paketov počas TCP spojenia v smere dnu.	K.
PktPerSesOut	Počet paketov počas TCP spojenia v smere von.	K.
TSesStart	Čas začiatku spojenia.	NK.
TSesEnd	Čas konca spojenia.	NK.
SessDuration	Doba trvania spojenia.	NK.
CntResendPktsIn	Počet znovu poslaných paketov v smere dnu.	NK.
CntResendPktsOut	Počet znovu poslaných paketov v smere von.	NK.
CntPktOutOfOrd	Počet paketov doručených mimo poradia.	
MedTdiff2Pkts	Medián času medzi 2 paketmi.	NK.
ModTdiff2Pkts	Modus času medzi 2 paketmi.	NK.
MeanTdiff2Pkts	Priemer časov medzi 2 paketmi.	NK.
SigTdiff2Pkts	Smerodajná odchýlka časov medzi 2 paketmi.	NK.
MedTdiff2PktsIn	Medián času medzi 2 paketmi v smere dnu.	NK.
ModTdiff2PktsIn	Modus času medzi 2 paketmi v smere dnu.	NK.
MeanTdiff2PktsIn	Priemer časov medzi 2 paketmi v smere dnu.	NK.
SigTdiff2PktsIn	Smerodajná odchýlka časov medzi 2 paketmi v smere dnu.	NK.
MedTdiff2PktsOut	Medián času medzi 2 paketmi v smere von.	NK.
ModTdiff2PktsOut	Modus času medzi 2 paketmi v smere von.	NK.
MeanTdiff2PktsOut	Priemer časov medzi 2 paketmi v smere von.	NK.
SigTdiff2PktsOut	Smerodajná odchýlka časov medzi 2 paketmi v smere von.	NK.
CntSYNIn	Celkový počet prijatých SYN paketov TCP.	K.
CntSYNOut	Celkový počet odoslaných SYN paketov TCP.	K.
CntACKIn	Celkový počet prijatých ACK paketov TCP.	K.
CntACKOut	Celkový počet odoslaných ACK paketov TCP.	K.
CntFINOut	Celkový počet odoslaných FIN paketov TCP.	K.
CntFINIn	Celkový počet prijatých FIN paketov TCP.	K.

Tabuľka 4.3: Dynamické metriky.

4.5 Lokalizačné metriky

Vlastnosťou metrík navrhnutých v rámci tejto kategórie je, že obsahujú vo väčšine prípadov v rámci spojenia statické vlastnosti. Tieto vlastnosti reprezentujú lokalizáciu strojov a pre komunikáciu použitých portov. Zoznam navrhnutých metrík je uvedený v tabuľke č. 4.4. Ak metrika rozlišuje smer komunikácie, tak je to explicitne uvedené v označení metriky (src/dst).

Označenie metriky	Stručný popis metriky	Poznámka
srcIP	IP adresa zdroja.	
srcIPInVlan	Príznak, či sa zdrojová IP adresa nachádza v rámci lokálnej siete.	
dstIP	Ip adresa cieľa.	
dstIPInVlan	Príznak, či sa cieľová IP adresa nachádza v rámci lokálnej siete.	
srcPort	Zdrojový port.	
dstPort	Cieľový port.	
srcMAC	Zdrojová MAC adresa.	
dstMAC	Cieľová MAC adresa.	

Tabuľka 4.4: Lokalizačné metriky.

4.6 Distribujúce metriky

U tejto kategórie metrík je charakteristickou vlastnosťou, že distribujú pakety alebo ich dĺžky do pevne stanoveného počtu intervalov a to za určitú dobu. Ďalšou charakteristickou vlastnosťou tejto kategórie metrík je ich vektorová forma reprezentácie. Vo všetkých uvedených metrikách tejto kategórie sa implicitne predpokladá, že uvažujú kontext skúmaného spojenia. Zoznam distribujúcich metrík je uvedený v tabuľke č. 4.5. Ak metrika rozlišuje prichádzajúce a odchádzajúce pakety, tak je to uvedené v označení metriky (in/out).

Označenie metriky	**Stručný popis metriky**	**Poznámka**
InPkt1s10i	Počty paketov za 1 sekundu distribuované do 10 intervalov v smere dnu.	
InPkt4s10i	Počty paketov za 4 sekúnd distribuované do 10 intervalov v smere dnu.	
InPkt8s10i	Počty paketov za 8 sekúnd distribuované do 10 intervalov v smere dnu.	
InPkt32s10i	Počty paketov za 32 sekúnd distribuované do 10 intervalov v smere dnu.	
InPkt64s10i	Počty paketov za 64 sekúnd distribuované do 10 intervalov v smere dnu.	
OutPkt1s10i	Počty paketov za 1 sekundu distribuované do 10 intervalov v smere von.	
OutPkt4s10i	Počty paketov za 4 sekúnd distribuované do 10 intervalov v smere von.	
OutPkt8s10i	Počty paketov za 8 sekúnd distribuované do 10 intervalov v smere von.	
OutPkt32s10i	Počty paketov za 32 sekúnd distribuované do 10 intervalov v smere von.	
OutPkt64s10i	Počty paketov za 64 sekúnd distribuované do 10 intervalov v smere von.	Prah môže byť stanovený aj inak.
InPkt1s10iTr4KB	Počty paketov za 1 sekundu distribuované do n intervalov podľa prahu 4 KB v smere dnu.	
InPkt4s10iTr4KB	Počty paketov za 4 sekúnd distribuované do n intervalov podľa prahu 4 KB v smere dnu.	
InPkt8s10iTr4KB	Počty paketov za 8 sekúnd distribuované do n intervalov podľa prahu 4 KB v smere dnu.	
InPkt32s10iTr4KB	Počty paketov za 32 sekúnd distribuované do n intervalov podľa prahu 4 KB v smere dnu.	
InPkt64s10iTr4KB	Počty paketov za 64 sekúnd distribuované do n intervalov podľa prahu 4 KB v smere dnu.	
OutPkt1s10iTr4KB	Počty paketov za 1 sekundu distribuované do n intervalov podľa prahu 4 KB v smere von.	
OutPkt4s10iTr4KB	Počty paketov za 4 sekúnd distribuované do n intervalov podľa prahu 4 KB v smere von.	
OutPkt8s10iTr4KB	Počty paketov za 8 sekúnd distribuované do n intervalov podľa prahu 4 KB v smere von.	
OutPkt32s10iTr4KB	Počty paketov za 32 sekúnd distribuované do n intervalov podľa prahu 4 KB v smere von.	
OutPkt64s10iTr4KB	Počty paketov za 64 sekúnd distribuované do n intervalov podľa prahu 4 KB v smere von.	
InPktLen1s10i	Dĺžky paketov za 1 sekundu distribuované do 10 intervalov v smere dnu.	
InPktLen4s10i	Dĺžky paketov za 4 sekúnd distribuované do 10 intervalov v smere dnu.	
InPktLen8s10i	Dĺžky paketov za 8 sekúnd distribuované do 10 intervalov v smere dnu.	
InPktLen32s10i	Dĺžky paketov za 32 sekúnd distribuované do 10 intervalov v smere dnu.	
InPktLen64s10i	Dĺžky paketov za 64 sekúnd distribuované do 10 intervalov v smere dnu.	
OutPktLen1s10i	Dĺžky paketov za 1 sekundu distribuované do 10 intervalov v smere von.	
OutPktLen4s10i	Dĺžky paketov za 4 sekúnd distribuované do 10 intervalov v smere von.	
OutPktLen8s10i	Dĺžky paketov za 8 sekúnd distribuované do 10 intervalov v smere von.	
OutPktLen32s10i	Dĺžky paketov za 32 sekúnd distribuované do 10 intervalov v smere von.	
OutPktLen64s10i	Dĺžky paketov za 64 sekúnd distribuované do 10 intervalov v smere von.	
InPkt64s20iTr2KB	Počty paketov za 64 sekúnd distribuované do n intervalov podľa stanoveného prahu 2 KB v smere dnu.	
InPkt64s20iTr1KB	Počty paketov za 64 sekúnd distribuované do n intervalov podľa stanoveného prahu 1 KB v smere dnu.	
OutPkt64s20iTr2KB	Počty paketov za 64 sekúnd distribuované do n intervalov podľa stanoveného prahu 2 KB v smere von.	
OutPkt64s20iTr1KB	Počty paketov za 64 sekúnd distribuované do n intervalov podľa stanoveného prahu 1 KB v smere von.	

Tabuľka 4.5: Distribujúce metriky.

4.7 Behaviorálne metriky

Charakteristikou metrík tejto kategórie je, že sa snažia interpretovať informáciu o priebehu spojenia v čase. Ich reprezentácia je tiež v mnohých prípadoch vektorová, no pre lepšiu štruktúrovanosť sú oddelené od predchádzajúcej kategórie. Zoznam metrík je uvedený v tabuľke č. 4.6 a 4.7. Ak metrika rozlišuje prichádzajúce a odchádzajúce pakety, tak je to explicitne uvedené v označení tejto metriky (in/out). Kontext analyzovaného spojenia sa implicitne v tejto kategórii metrík neuvažuje. V situáciách kde sa uvažuje, je to explicitne uvedené v poznámke skratkou K.

Označenie metriky	**Stručný popis metriky**	**Poznámka**
cntOfOldFlows	Počet spoločných tokov uzlov spojenia pred aktuálnym spojením.	Vyžaduje analýzu všetkých spoločných komunikácií uzlov spojenia. Časové ohraničenie 5 minút.
cntOfNewFlows	Počet nových spojení po začatí skúmaného spojenia.	Vyžaduje analýzu všetkých spoločných komunikácií uzlov spojenia. Časové ohraničenie 5 minút. Zohľadňuje možnosť komunikácie na backdoor porte.
corClosed	Príznak korektného ukončenia TCP spojenia.	Pomocou trojcestného endshaku.
polynomIndexes3ordIn	Aproximácia prichádzajúcej komunikácie polynómom 3. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes3ordOut	Aproximácia odchádzajúcej komunikácie polynómom 3. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes5ordIn	Aproximácia prichádzajúcej komunikácie polynómom 5. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes5ordOut	Aproximácia odchádzajúcej komunikácie polynómom 5. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes8ordIn	Aproximácia prichádzajúcej komunikácie polynómom 8. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes8ordOut	Aproximácia odchádzajúcej komunikácie polynómom 8. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes10ordIn	Aproximácia prichádzajúcej komunikácie polynómom 10. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes10ordOut	Aproximácia odchádzajúcej komunikácie polynómom 10. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes13ordIn	Aproximácia prichádzajúcej komunikácie polynómom 13. rádu.	Osou x je index paketu v rámci spojenia.
polynomIndexes13ordOut	Aproximácia odchádzajúcej komunikácie polynómom 13. rádu.	Osou x je index paketu v rámci spojenia.
polynomTimestamps3ordIn	Aproximácia prichádzajúcej komunikácie polynómom 3. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps3ordOut	Aproximácia odchádzajúcej komunikácie polynómom 3. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps5ordIn	Aproximácia prichádzajúcej komunikácie polynómom 5. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps5ordOut	Aproximácia odchádzajúcej komunikácie polynómom 5. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps8ordIn	Aproximácia prichádzajúcej komunikácie polynómom 8. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps8ordOut	Aproximácia odchádzajúcej komunikácie polynómom 8. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps10ordIn	Aproximácia prichádzajúcej komunikácie polynómom 10. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps10ordOut	Aproximácia odchádzajúcej komunikácie polynómom 10. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps13ordIn	Aproximácia prichádzajúcej komunikácie polynómom 13. rádu.	Osou x je čas paketu v rámci spojenia.
polynomTimestamps13ordOut	Aproximácia odchádzajúcej komunikácie polynómom 13. rádu.	Osou x je čas paketu v rámci spojenia.

Tabuľka 4.6: Behaviorálne metriky (časť 1).

Označenie metriky	Stručný popis metriky	Poznámka
fourCoefsGonModulIn	Koeficienty Fourierovej rady v goniometrickej reprezentácii (modul). Smer prichádzajúca komunikácia.	
fourCoefsGonAngleIn	Koeficienty Fourierovej rady v goniometrickej reprezentácii (uhol). Smer prichádzajúca komunikácia.	
fourCoefsGonModulOut	Koeficienty Fourierovej rady v goniometrickej reprezentácii (modul). Smer odchádzajúca komunikácia.	
fourCoefsGonAngleOut	Koeficienty Fourierovej rady v goniometrickej reprezentácii (uhol). Smer odchádzajúca komunikácia.	
intervalsPortsSig	Smerodajná odchýlka časových intervalov medzi spojeniami prebiehajúcich na rovnakých portoch aj IP. Uvažuje začiatky spojení na určenie intervalov.	K. V laboratórnych podmienkach nevyužiteľné.
intervalsPortsSig2	Smerodajná odchýlka časových intervalov medzi spojeniami prebiehajúcich na rovnakých portoch aj IP. Uvažuje začiatky aj konce spojení na určenie intervalov.	K. V laboratórnych podmienkach nevyužiteľné.
intervalsIPsSig	Smerodajná odchýlka časových intervalov medzi spojeniami prebiehajúcich na rovnakých adresách IP. Uvažuje začiatky spojení na určenie intervalov.	K. V laboratórnych podmienkach nevyužiteľné.
intervalsIPsSig2	Smerodajná odchýlka časových intervalov medzi spojeniami prebiehajúcich na rovnakých adresách IP. Uvažuje začiatky aj konce spojení na určenie intervalov.	K. V laboratórnych podmienkach nevyužiteľné.
gaussProds1In	Normalizované produkty vstupnej komunikácie s 1 gaussovou krivkou.	
gaussProds2In	Normalizované produkty vstupnej komunikácie s 2 gaussovými krivkami.	
gaussProds4In	Normalizované produkty vstupnej komunikácie s 4 gaussovými krivkami.	
gaussProds8In	Normalizované produkty vstupnej komunikácie s 8 gaussovými krivkami.	
gaussProds1Out	Normalizované produkty výstupnej komunikácie s 1 gaussovou krivkou.	
gaussProds2Out	Normalizované produkty výstupnej komunikácie s 2 gaussovými krivkami.	
gaussProds4Out	Normalizované produkty výstupnej komunikácie so 4 gaussovými krivkami.	
gaussProds8Out	Normalizované produkty výstupnej komunikácie s 8 gaussovými krivkami.	
gaussProds1All	Normalizované produkty zlúčenej komunikácie s 1 gaussovou krivkou.	
gaussProds2All	Normalizované produkty zlúčenej komunikácie s 2 gaussovými krivkami.	
gaussProds4All	Normalizované produkty zlúčenej komunikácie so 4 gaussovými krivkami.	
gaussProds8All	Normalizované produkty zlúčenej komunikácie s 8 gaussovými krivkami.	

Tabuľka 4.7: Behaviorálne metriky (časť 2).

4.8 Problém priepustnosti a vzdialenosti určitých metrík

Jedným z problémov pri vlastnom návrhu metrík je použitie sieťových kariet s rôznou rýchlosťou prenosu. Jedná sa napríklad o rozdiel pri spracovávaní dát na 100MB/s linke prípadne 1GB/s linke. Ďalší vplyv môže mať aj spomalenie na úrovni PC zbernice spôsobenej danou architektúrou konkrétneho PC. Môže tu dôjsť k extrakcii rôznych hodnôt niektorých metrík pre 2 identické simulácie TCP komunikácie. Tento problém rovnako môže zapríčiňovať aj rôzna geografická vzdialenosť komunikujúcich strán. Väčší počet smerovacích a prepínacích prvkov po ceste zapríčiní rôzne prenosové latencie. K ovplyvneniu týmto problémom dochádza najmä u metrík pracujúcich s časom:

- **väčšina dynamických** – počty prenesených bajtov alebo paketov za sekundu, čas začiatku alebo konca spojenia a ďalšie metriky využívajúce čas medzi doručením dvoch paketov,
- **väčšina distribučných** – okrem tých, ktoré využívajú stanovený prah a podľa neho uskutočňujú distribúciu,
- **niektoré behaviorálne** – aproximácie polynómami s využitím času, počet nových spojení po začatí skúmaného spojenia a počet spoločných tokov uzlov spojenia pred skúmaným spojením.

Preto vznikla snaha využiť také aproximačné techniky, ktoré sú na čase nezávislé a pracujú s indexami jednotlivých paketov. Konkrétne boli uplatnené u aproximácií priebehov komunikácií polynómami s využitím indexov paketov, ďalej u aproximácií Fourierovými radami a tiež u normalizovaných produktoch priebehu komunikácií s gaussovými krivkami.

4.9 Aproximácia TCP komunikácií polynómami

Pri návrhu metrík patriacich do rodiny aproximácií polynómami bolo vychádzané z [18]. Vzhľadom na to, že jednotlivé komunikácie môžu obsahovať rôzne počty paketov, pripadajú v úvahu len metódy aproximácií krivkami. Pri interpolácii krivkami metódy produkujú variabilný počet koeficientov, čo je pre náš prípad nevhodné. Pre navrhnutie týchto metrík bola použitá metóda najmenších štvorcov. Metódu najmenších štvorcov je možné použiť na aproximáciu pomocou všeobecnej krivky.

4.9.1 Formulácia problému

Nasledujúce úvahy sú čerpané z [18]. Sú dané body x_i, $i = 0, \ldots, n$ a funkčné hodnoty v nich y_i. Ďalej sú dané funkcie φ_i, $i = 0, \ldots, m$, $m < n$ (pre priamku by to boli funkcie $\varphi_0(x) = 1$ a $\varphi_1(x) = x$, pre parabolu by k nim navyše pribudla funkcia $\varphi_2(x) = x^2$). Medzi všetkými funkciami tvaru

$$P_m(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \ldots + c_m\varphi_m(x), \tag{4.1}$$

kde $c_0, \ldots, c_m$ sú reálne čísla, hľadáme takú funkciu, pre ktorú kvadratická odchýlka

$$\rho^2(c_0, \ldots, c_m) = \sum_{i=0}^{n} (y_i - P_m(x_i))^2 \tag{4.2}$$

nadobúda minimálnu hodnotu. Takúto funkciu nazývame najlepšou aproximáciou experimentálnych dát $y_0, ..., y_n$ v danej triede funkcií v zmysle metódy najmenších štvorcov. Kvadratická odchýlka

$$\rho^2 = \sum_{i=0}^{n} (y_i - c_0\varphi_0(x_i) - c_1\varphi_1(x_i) - \ldots - c_m\varphi_m(x_i))^2 \tag{4.3}$$

je minimálna v tom bode $(c_0, \ldots, c_m)$, v ktorom sú splnené rovnice

$$\frac{\partial(\rho^2)}{\partial c_j} = \frac{\partial}{\partial c_j} \left[\sum_{i=0}^{n} (y_i - c_0\varphi_0(x_i) - c_1\varphi_1(x_i) - \ldots - c_m\varphi_m(x_i))^2 \right] = 0, \qquad j = 0, \ldots, m. \tag{4.4}$$

Zderivovaním dostaneme

$$\sum_{i=0}^{n} 2(y_i - c_0\varphi_0(x_i) - c_1\varphi_1(x_i) - \ldots - c_m\varphi_m(x_i))(-\varphi_j(x_i)) = 0, \qquad j = 0, \ldots, m. \tag{4.5}$$

Rovnice vydelíme -2 a rozdelíme na jednotlivé sumy:

$$\sum_{i=0}^{n} y_i\varphi_j(x_i) - \sum_{i=0}^{n} c_0\varphi_0(x_i)\varphi_j(x_i) - \ldots - \sum_{i=0}^{n} c_m\varphi_m(x_i)\varphi_j(x_i) = 0, \qquad j = 0, \ldots, m. \tag{4.6}$$

Z každej sumy môžeme vyňať odpovedajúci koeficient c_k. Jednoduchou úpravou dostávame normálne rovnice pre neznáme $c_0, \ldots, c_m$:

$$c_0 \sum_{i=0}^{n} \varphi_0(x_i)\varphi_j(x_i) + \ldots + c_m \sum_{i=0}^{n} \varphi_m(x_i)\varphi_j(x_i) = \sum_{i=0}^{n} y_i\varphi_j(x_i), \qquad j = 0, \ldots, m. \tag{4.7}$$

Táto sústava po rozpísaní vyzerá nasledovne:

$$\begin{aligned}
c_0 \sum_{i=0}^{n} \varphi_0^2(x_i) + c_1 \sum_{i=0}^{n} \varphi_1(x_i)\varphi_0(x_i) + \ldots + c_m \sum_{i=0}^{n} \varphi_m(x_i)\varphi_0(x_i) &= \sum_{i=0}^{n} y_i\varphi_0(x_i) \\
c_0 \sum_{i=0}^{n} \varphi_0(x_i)\varphi_1(x_i) + c_1 \sum_{i=0}^{n} \varphi_1^2(x_i) + \ldots + c_m \sum_{i=0}^{n} \varphi_m(x_i)\varphi_1(x_i) &= \sum_{i=0}^{n} y_i\varphi_1(x_i) \\
\ddots \quad \vdots \\
c_0 \sum_{i=0}^{n} \varphi_0(x_i)\varphi_m(x_i) + c_1 \sum_{i=0}^{n} \varphi_1(x_i)\varphi_m(x_i) + \ldots + c_m \sum_{i=0}^{n} \varphi_m^2(x_i) &= \sum_{i=0}^{n} y_i\varphi_m(x_i)
\end{aligned}$$

Riešením sústavy týchto rovníc dostávame koeficienty aproximujúcej krivky $c_0, \ldots, c_m$, ktoré najlepšie aproximujú vstupný súbor dát. V rámci tejto práce boli použité len aproximácie polynomiálnymi krivkami.

Pre účely tejto práce boli zvolené polynómy rôznych stupňov uvedených v predchádzajúcom texte. Boli použité dve varianty aproximácie. V prvej variante bolo ako os x použité poradie daného paketu a v druhej variante bol ako os x použitý relatívny čas zachytenia paketu od začiatku komunikácie.

Vzhľadom na nízke počty paketov jednotlivých spojení v testovacích dátach sa predpokladá, že lepšie výsledky prinesú aproximácie polynómami nižších rádov. Metriky s aproximáciami polynómami vyšších rádov boli pridané z dôvodu ďalších potenciálnych testovaní v reálnych sieťach, kde aktuálna povaha použitých testovacích dát nemusí byť významná.

Kapitola 5

Implementácia extrakcie metrík

Pre implementačnú časť extrakcie spojení a metrík bol zvolený programovací jazyk Python[1]. Dáta reprezentujúce ako útoky, tak aj legitímne komunikácie boli poskytnuté spoluriešiteľmi projektu MPO AIPS. Tieto dáta sú dostupné cez PostgreSQL[2] databázový server *http://buslab-11.fit.vutbr.cz/*. Jedná sa o informácie získané aplikáciou TCPdump. Jednotlivé útoky boli uskutočňované prostredníctvom aplikácie metasploit[3], kde cieľom útoku bola virtuálna stanica bežiaca pod honeypotom Argos.

5.1 Vývojové prostredie

Všetká programová časť tejto práce bola implementovaná pod operačným systémom Linux Fedora 15[4] s jadrom 2.6.42. Bola použitá verzia 2.7.1 interpretu jazyka Python. Pre kolaboračné a zálohovacie účely bol použitý repozitárový systém GIT[5].

5.2 Argos a metasploit

Aplikácia metasploit obsahuje databázu útokov využívajúcich rôzne doposiaľ známe zraniteľnosti rozličných služieb. Prioritným cieľom metasploitu je pomáhať bezpečnostným analytikom a IT profesionálom identifikovať prítomnosť zraniteľností na skúmaných systémoch. Jeho ďalšími vlastnosťami sú možnosť exploitácie služieb, audit hesiel, skenovanie webových aplikácií a sociálne inžinierstvo [11]. Existujú rôzne edície tohto software, no v rámci simulácie útokov bola použitá voľne dostupná verzia 3.7.2.

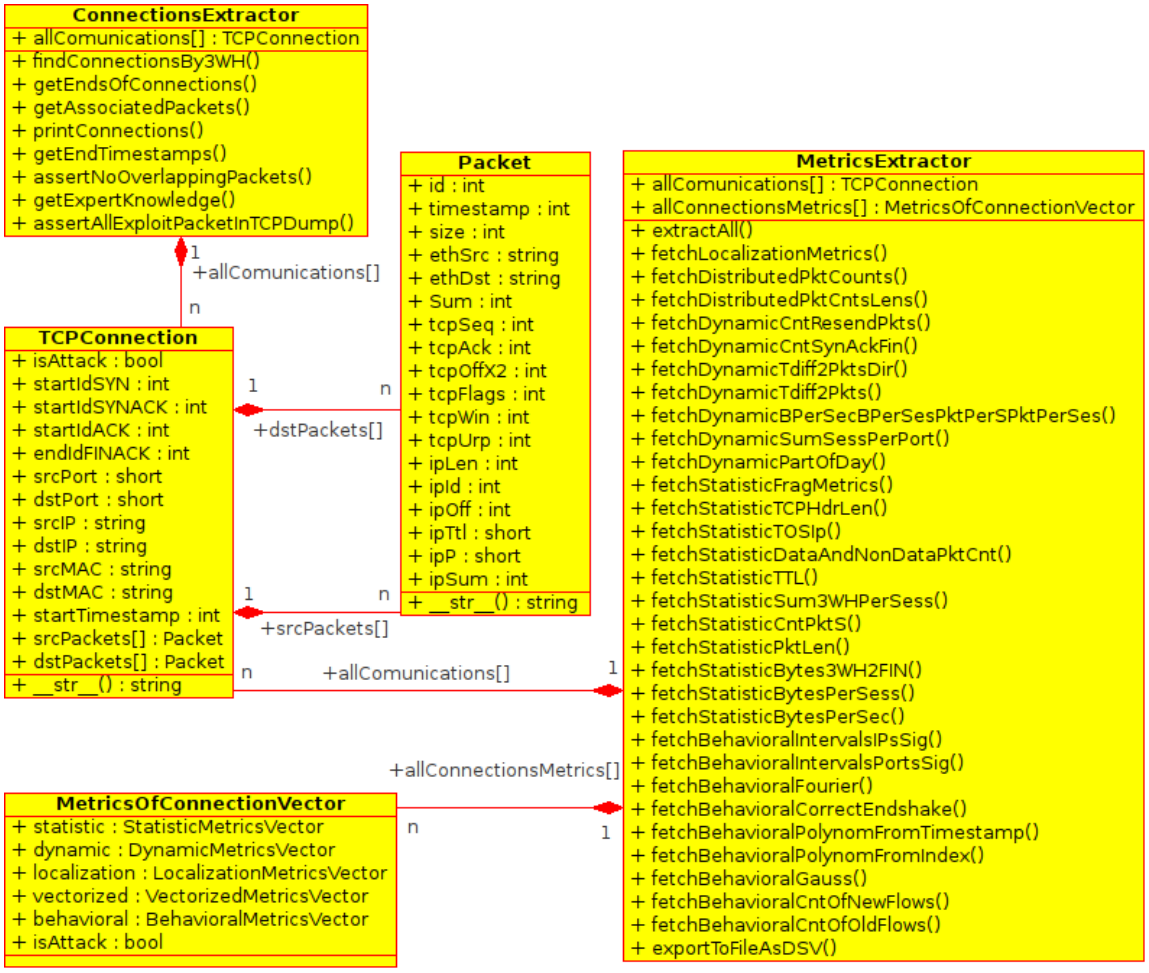
Pri simulácií útoku na klientskú stanicu bežiacu pod honeypotom Argos, dôjde k detekcii tohto útoku pričom sa zaznamenávajú obsahy jednotlivých registrov procesora, relevantné pamäťové bloky a konkrétny paket, ktorý tento útok spôsobil. Je to možné pomocou techník ako *dynamic taint analysis* popísaných v kapitole 2.3.5. Prostredníctvom týchto znalostí je možné identifikovať spojenie, ktoré útok spôsobilo. Pre naše účely je relevantná identifikácia spojenia, ktoré spôsobilo napadnutie systému, a preto obsahy registrov a pamäťových blokov napadnutého procesu nebudú ďalej uvažované.

Detekcia *buffer overflow* útoku pomocou honeypotu Argos jedinou expertnou znalosťou hovoriacou, že došlo k útoku.

[1] Interpretovaný, interaktívny programovací jazyk. http://python.org/.
[2] Voľne šíriteľný objektovo-relačný databázový systém. http://www.postgresql.org/.
[3] Viď sekcia 5.2. http://metasploit.com/.
[4] Linuxová distribúcia založená na balíčkovacom systéme RPM. http://fedoraproject.org/wiki/Releases/15.
[5] Distribuovaný systém riadenia revízií. http://git-scm.com/.

5.3 Implementácia extrakcie spojení a ich metrík

Ako objekt zvolený pre analýzu a klasifikáciu bolo stanovené TCP spojenie, a preto bolo potrebné najskôr identifikovať hranice TCP spojení, kde sa začiatok detekoval pomocou *3-way handshaku* a koniec buď pomocou *3-way endshaku* alebo sa vôbec v komunikácii nevyskytol. Preto pri identifikácii paketov patriacich do daného spojenia bez nájdeného *3-way endshaku* sa uvažovali všetky relevantné pakety prijaté alebo odoslané do 2 minút od začiatku spojenia. Ďalej sa spravila samotná extrakcia metrík pre jednotlivé spojenia kde sa v niektorých prípadoch bral do úvahy aj kontext spojení (iné predchádzajúce, nasledujúce alebo paralelné spojenia). V rámci tejto práce sa naimplementovali všetky navrhnuté metriky každej kategórie. Diagram tried pre túto extrakciu je možné vidieť na obrázku č. 5.1 a obrázku č. 5.2. Popis funkcionality jednotlivých tried a zároveň metódy sprostredkujúce danú funkcionalitu bude uvedený v ďalších podsekciách.



Obrázok 5.1: Diagram tried extrakcie metrík (1. časť).

Obrázok 5.2: Diagram tried extrakcie metrík (2. časť).

5.3.1 Trieda ConnectionsExtractor

Trieda slúži na extrakciu informácií o spojeniach a extrakciu paketov asociovaných s danými spojeniami. Sú tu zapúzdrené nízkoúrovňové operácie pracujúce nad databázou pomocou databázového adaptéra pre Python - psycopg2[6]. Výstupom činnosti inštancie tejto triedy je zoznam všetkých spojení v atribúte allCommunications. Trieda ConnectionExtractor obsahuje nasledujúce metódy:

- *findConnectionsBy3WH* – nájde začiatky všetkých TCP komunikácií v databáze a naplní nimi svoj atribút typu zoznam *allComunications* nad objektami typu TCP-Connection,

[6]PostgreSQL adaptér pre python. http://initd.org/psycopg/.

- *getEndsOfConnections* – nájde konce všetkých komunikácií, ktoré má interne uložené a aktualizuje zoznam *allComunications*,
- *printConnections* – na štandardný výstup vypíše informácie o všetkých interne uložených spojeniach,
- *getEndTimestamps* – pre všetky interne uložené spojenia v zozname *allComunications* nájde pakety s najvyššími časovými razítkami a uloží ich v tomto zozname,
- *getAssociatedPackets* – pre všetky interne uložené spojenia v zozname *allComunications* nájde asociované pakety a uloží ich do jednotlivých objektov tohto zoznamu. Asociované pakety obsahujú aj pakety z trojcestného handshaku aj endshaku. Ak trojcestný endshake nebol nájdený, tak v tom prípade, extrahuje pakety na rovnakých adresách IP a rovnakých portoch po do doby 2 minút od začiatku spojenia.
- *assertNoOverlappingPackets* – overí tvrdenie, že všetky pakety každého spojenia boli extrahované správne a žiadny paket sa výskytom neprekrýva z paketmi iného spojenia,
- *getExpertKnowledge* – pre každé spojenie uložené v internom zozname *allComunications* zistí, či sa jedná o útok alebo nie. Ďalej túto informáciu expertnej znalosti zapíše do príslušného objektu tohto zoznamu. Tieto údaje sú získané z tabuľky *exploitpackets,* ktorú generoval honeypot Argos. Táto tabuľka obsahuje dáta L2 rámcov, ktoré spôsobili útok *buffer overflow*. Ku detekcii útoku dochádza hľadaním podreťazca užitočných dát TCP paketov analyzovaného objektu *TCPConnection* v dátach tabuľky *exploitpackets,*
- *assertAllExploitPacketInTCPDump* – overuje tvrdenie, že všetky útoky, ktoré zachytil Argos sa nachádzajú v databáze dát z TCPdump,

5.3.2 Trieda MetricsExtractor

Trieda slúži na extrakciu jednotlivých metrík v rámci rozdelenia podľa kategórií. Vstupom a zároveň parametrom konštruktoru je zoznam spojení vytvorený pomocou inštancie triedy *ConnectionsExtractor*. Sú tu implementované metódy, ktoré pre každé TCP spojenie extrahujú príslušné metriky. Táto trieda uskutočňuje tiež export extrahovaných metrík všetkých spojení do súboru vo forme DSV[7]. Výstupom činnosti inštancie tejto triedy je zoznam vektorov metrík pre jednotlivé spojenia uložený v atribúte *allConnectionsMetrics*. Trieda *MetricsExtractor* obsahuje nasledujúce metódy:

- *extractAll* – extrakcia je centralizovaná v tejto metóde - pre každé spojenie v zozname spojení volá príslušné metódy určené na extrakciu jednotlivých metrík a tieto extrahované metriky ukladá do svojho interného zoznamu *allConnectionsMetrics*, ktorý obsahuje objekty typu *MetricsOfConnectionVector*.
- *fetchLocalizationMetrics* – extrahuje všetky lokalizačné metriky,
- *fetchDistributedPktCounts* – extrahuje vektor počtu paketov v smere danom parametrom *direction* za čas daný parametrom *time*. Jednotlivé prvky vektoru sú oddelené po naakumulovaní prahovej hodnoty prenesených dát danej parametrom *threshold*. Vektor má stanovenú veľkosť na 20 prvkov,

[7] Delimiter separated values.

- *fetchDistributedPktCntsLens* – extrahuje vektor počtov a veľkostí paketov analyzovaného spojenia rozdelených do 10 intervalov podľa časového úseku braného v úvahu, ktorý je daný parametrom *duration.* Smer komunikácie je rozlíšený parametrom *direction*,

- *fetchDynamicCntResendPkts* – extrahuje počet znova poslaných paketov analyzovaného spojenia v smere danom parametrom *direction.* Znova poslané pakety identifikuje podľa rovnakého sekvenčného čísla v TCP hlavičke paketov,

- *fetchDynamicCntSynAckFin* – extrahuje počty TCP paketov typu SYN, ACK, FIN počas doby trvania analyzovaného spojenia v smere špecifikovanom parametrom *direction.* Uvažuje kontext spojenia,

- *fetchDynamicTdiff2PktsDir* – extrahuje medián, modus, priemer a smerodajnú odchýlku času medzi dvomi paketmi v analyzovanom spojení s ohľadom na smer daný parametrom *direction*,

- *fetchDynamicTdiff2Pkts* – extrahuje medián, modus, priemer a smerodajnú odchýlku času medzi dvomi paketmi v analyzovanom spojení bez ohľadu na smer,

- *fetchDynamicBPerSecBPerSesPktPerSPktPerSes* – extrahuje rýchlosť v bajtoch za sekundu, celkový počet prenesených bajtov, rýchlosť v paketoch za sekundu a celkový počet prenesených paketov za dobu analyzovaného TCP spojenia v smere špecifikovanom parametrom *direction.* Uvažuje kontext spojenia,

- *fetchDynamicSumSessPerPort* – extrahuje počet spojení s rovnakou službou (cieľovým portom) ako analyzované spojenie v dobe +- 5 minút od začiatku komunikácie. Berie v úvahu kontext spojenia,

- *fetchDynamicPartOfDay* – uvažuje deň rozdelený na 1000 intervalov (inšpiráciou bol itime[8]). Vracia poradové číslo intervalu, v ktorom sa vyskytoval začiatok analyzovaného spojenia,

- *fetchStatisticFragMetrics* – extrahuje metriky súvisiace s fragmentáciou paketov spojenia: počet fragmentovaných paketov, počet nefragmentovaných paketov a ich pomer,

- *fetchStatisticTCPHdrLen* – extrahuje štatistické parametre dĺžky hlavičky TCP spojenia ako sú smerodajná odchýlka, priemer, suma, modus a medián,

- *fetchStatisticTOSIp* – extrahuje modus a medán poľa TOS hlavičky IP paketov analyzovaného spojenia,

- *fetchStatisticDataAndNonDataPktCnt* – extrahuje počet dátových a nedátových paketov analyzovaného spojenia v smere danom parametrom *direction*,

- *fetchStatisticTTL* – extrahuje štatistické parametre modus, medián, smerodajnú odchýlku, priemer a sumu poľa TTL hlavičiek IP paketov analyzovaného spojenia v smere danom parametrom *direction*,

- *fetchStatisticSum3WHPerSess* – extrahuje počet TCP paketov typu SYN a SYNACK v analyzovanom spojení,

[8] Internet time. http://www.timeanddate.com/time/internettime.html.

- *fetchStatisticCntPktS* – extrahuje počet paketov za sekundu v analyzovanom spojení a smere danom parametrom *direction*. Reprezentuje rýchlosť spojenia v danom smere,
- *fetchStatisticPktLen* – extrahuje štatistické parametre ako medián, modus, priemer, smerodajná odchýlka a suma dĺžok paketov v analyzovanom spojení s ohľadom na smer daný parametrom *direction*,
- *fetchStatisticBytes3WH2FIN* – extrahuje celkový počet prenesených bajtov od začiatku spojenia až do konca (vrátane paketov na zostavenie a ukončenie spojenia),
- *fetchStatisticBytesPerSess* – extrahuje počet prenesených bajtov od začiatku spojenia až do konca (bez ohľadu na zostavenie a ukončenie spojenia),
- *fetchStatisticBytesPerSec* – extrahuje počet bajtov za sekundu v smere špecifikovanom parametrom *direction,*
- *fetchBehavioralIntervalsIPsSig* – extrahuje štandardnú odchýlku časových intervalov medzi spojeniami komunikujúcich na rovnakých IP adresách ako má analyzované spojenie. Uvažuje kontext spojenia,
- *fetchBehavioralIntervalsPortsSig* – extrahuje štandardnú odchýlku časových intervalov medzi spojeniami komunikujúcich na rovnakých portoch ako má analyzované spojenie. Uvažuje kontext spojenia,
- fetchBehavioralFourier – extrahuje vektor komplexných čísel určujúcich koeficienty Fourierovej rady aproximujúcej priebeh analyzovaného spojenia v smere danom parametrom *direction*. Uskutočňuje aj prepočet na goniometrickú reprezentáciu komplexných čísel. Vracia prvých 20 koeficientov Fourierovej rady,
- *fetchBehavioralCorrectEndshake* – zistí, či bolo analyzované spojenie korektne ukončené,
- *fetchBehavioralPolynomFromTimestamp* – nájde koeficienty polynómu, ktorý metódou najmenších štvorcov najlepšie aproximuje pakety analyzovaného spojenia v čase so špecifikovaným smerom. Rád polynómu je daný parametrom *order,*
- *fetchBehavioralPolynomFromIndex* – nájde koeficienty polynómu, ktorý metódou najmenších štvorcom najlepšie aproximuje pakety analyzovaného spojenia v diskrétnej ose indexov so špecifikovaným smerom. Rád polynómu je daný parametrom *order,*
- *fetchBehavioralGauss* – extrahuje vektor normalizovaných produktov s diskrétnymi gaussovými funkciami a diskrétnou funkciou dĺžiek paketov analyzovaného spojenia s ohľadom na smer daný parametrom *direction*. Počet susediacich gaussových funkcií, s ktorých sa získava produkt je daný parametrom *productsCnt*,
- *fetchBehavioralCntOfNewFlows* – extrahuje počet nových TCP spojení s rovnakými IP adresami ako analyzované spojenie vzniknutých od začiatku tohto spojenia do 5 minút od začiatku tohto spojenia. Uvažuje kontext spojenia,
- *fetchBehavioralCntOfOldFlows* – extrahuje počet TCP spojení s rovnakými IP adresami ako analyzované spojenie vzniknutých maximálne 5 minút pred začiatkom tohto spojenia do začiatku tohto spojenia. Uvažuje kontext spojenia,

- exportToFileAsDSV – uskutočňuje exportovanie vnútorného zoznamu extrahovaných metrík do DSV súboru. Export je uskutočňovaný paralelne v dvoch formátoch rozlišujúcich prístup k vektorovým atribútom.

5.3.3 Trieda TCPConnection

Trieda slúži ako štruktúra na uloženie informácií o TCP spojení extrahované z databázy alebo získané iným nepriamym spôsobom. Najdôležitejšie atribúty začiatok a koniec TCP spojenia a tiež zoznamy prichádzajúcich (*dstPackets*) a odchádzajúcich (*srcPackets*) paketov danej inštancie spojenia (z pohľadu strany, ktorá spojenie iniciovala). Tieto zoznamy obsahujú objekty typu *Packet*. Trieda ďalej obsahuje metódu na konverziu obsahu inštancie do textového reťazca využívanú najmä pre tlačenie na štandardný výstup pri extrakcii.

5.3.4 Trieda Packet

Reprezentuje TCP paket so všetkými náležitosťami, ktoré sú priamo extrahovateľné z poskytnutej databázy naplnenej aplikáciou TCPdump. Najdôležitejšími atribútmi tejto triedy sú čas výskytu (*timestamp*) a tiež veľkosť paketu (*size*). Trieda ďalej obsahuje metódu na konverziu obsahu inštancie do textového reťazca využívanú najmä pre tlačenie na štandardný výstup pri extrakcii.

5.3.5 Ďalšie triedy

Pre lepšiu štruktúrovanosť vektoru extrahovaných metrík bolo zavedené rovnaké členenie metrík ako pri ich návrhu. Trieda združujúca všetky podkategórie, nesie názov **MetricsOfConnectionVector.** Reprezentuje všetky extrahované metriky pre jedno TCP spojenie. Skladá sa z atribútov, ktoré tieto metriky kategorizujú do podvektorov a nasledujúce podvektory sú reprezentované týmito triedami:

- **LocalizationMetricsVector** – reprezentuje lokalizačné metriky TCP spojenia,
- **BehavioralMetricsVector** – reprezentuje behaviorálne metriky TCP spojenia,
- **StatisticMetricsVector** – reprezentuje štatistické metriky TCP spojenia,
- **DynamicMetricsVector** – reprezentuje dynamické metriky TCP spojenia,
- **DistributedMetricsVector** – reprezentuje distribujúce metriky TCP spojenia**.**

Na účely správneho exportu desatinných čísel samotných alebo vo forme vektorov boli zavedené špeciálne triedy **specialFloat** a **specialFloatList** zapúzdrujúce preddefinovaný spôsob tohoto exportu, ktorý akceptuje dolovací nástroj. Tieto triedy boli navrhnuté a naimplementované až v čase kedy použitý dolovací nástroj detekoval chýbajúce hodnoty niektorých atribútov pričom sa predpokladalo, že žiadne chýbajúce hodnoty výstupný DSV súbor neobsahuje. Bolo to z dôvodu, že čísla v exponenciálnom tvare dolovací nástroj RapidMiner implicitne ignoroval a na miesto nich nechával prázdne hodnoty.

5.4 Konfiguračné nastavenia

Implementovaný nástroj na extrakciu metrík využíva nastavenia uložené v konfiguračnom súbore *Config.py*. Tieto nastavenia sa týkajú najme formy exportu do DSV formátu, pripojenia k databáze a siete, v ktorej sa nasadený honeypot nachádza.

Pre správnu konfiguráciu databázy je nutné aby v uvedenom súbore boli vyplnené názov databáze (*DATABASE*), prihlasovacie meno (*USER*) a heslo (*PASSWORD*) k prístupu do tejto databáze, ktorá sa nachádza na stroji s adresou špecifikovanou v ďalšom parametri (*HOST*). Nachádza sa tu tiež parameter s portom (*PORT*), na ktorom databázový server beží.

Čo sa týka exportu extrahovaných metrík do DSV formátu, tak najpodstatnejším parametrom je oddeľovač použitý na oddelenie jednotlivých metrík (*DSV_DELIM*). Ďalší parameter presne špecifikuje spôsob zaobchádzania s vektorovými atribútmi pri exporte (VECTORS_AS_LIST). Nadobúda hodnôt *true* a *false*, kde pri *true* sa vektorové atribúty exportujú ako pythonovské zoznamy zavolaním metódy realizujúcej implicitnú konverziu pythonovských zoznamov na textové reťazce, čím vznikne jeden atribút na jeden vektor. Pri hodnote *false* sa každá položka vektoru exportuje ako samostatný atribút (metrika).

Nastavenia umiestnenia honeypotu obsahujú dve položky: adresu siete (NETWORK) a sieťovú masku (NETWORK_MASK). Tieto nastavenia siete sa využívajú pri extrakcii lokalizačných metrík *srcIpInVlan* a *dstIpInVlan,* ktoré určujú či sa zdrojová resp. cieľová adresa nachádza v lokálnej sieti honeypotu. V budúcnosti toto nastavenie môže byť využité aj na ďalšie účely.

5.5 Popis behu aplikácie

Princíp činnosti skriptu na extrakciu metrík je nasledovný: Vytvorí sa inštancia triedy *ConnectionsExtractor*, ktorá najskôr identifikuje spojenia podľa 3WH a následne sa k týmto spojeniam pokúsi nájsť korektné ukončenia prostredníctvom 3WE. Potom pre všetky spojenia nájde asociované pakety a uloží ich do interných zoznamov jednotlivých spojení. Po asociovaní paketov nájde pre každé spojenie koncový čas, ktorý opäť uloží do jednotlivých inštancií spojení (atribút *endTimestamp*). Následne sa zavolá metóda objektu typu *ConnectionsExtractor*, ktorá pre každé spojenie zistí, či sa jedná o útok alebo o legitímnu komunikáciu a túto informáciu do príslušného objektu TCP spojenia uloží. Po extrakcii spojení sa voliteľne volá metóda, ktorá ich vypíše ich textové reprezentácie. Tieto extrahované spojenia z databázy si uloží do svojho atribútu *allCommunications*.

Potom sa vytvorí inštancia triedy *MetricsExtractor*, ktorá dostane v konštruktore referenciu na zoznam všetkých spojení od inštancie triedy *ConnectionsExtractor*. Ďalej sa zavolá metóda *extractAll*, ktorá extrahuje všetky metriky pre každé spojenie takým spôsobom, že prejde svoj interný zoznam spojení a pre každé spojenia vytvorí inštanciu triedy *MetricsOfConnectionVector*. A do tejto inštancie sa postupne pridávajú jednotlivé extrahované metriky. Po skončení extrakcie všetkých metrík sa tento objekt uloží do interného zoznamu *allConnectionsMetrics* triedy *MetricsExtractor*. Tieto metriky je potrebné uložiť do výstupného súboru, čo je realizované následným zavolaním metódy *exportToFileAsDSV* objektu triedy *MetricsExtractor*.

5.6 Generovanie výstupu

Použitý formát výstupu je zamýšľaný pre využitie dolovacími nástrojmi ako RapidMiner. Prvý riadok výstupného DSV súboru je popis daného stĺpca, ktorý slúži len pre štruktúrovanosť a prehľadnosť kategórií jednotlivých metrík. U jednotlivých metrikách príslušný stĺpec tohoto riadku informuje o kategórii metriky. Zároveň tento riadok určuje, ktorý stĺpec tvorí triedu spojenia (útok alebo legitímna komunikácia). Pre dolovací nástroj je

tento riadok len komentárom.

Ďalší riadok obsahuje názvy jednotlivých metrík a určuje tiež názov stĺpca s identifikáciou spojenia (*id*). Návestie triedy spojenia je v špeciálnom stĺpci *label*.

Po týchto dvoch riadkoch nasledujú DSV záznamy jednotlivých spojení. Jednotlivé kategórie metrík tvoria v tomto výstupnom súbore vždy súvislé úseky a ich vzájomné poradie je možné meniť, prípadne podľa potreby vyexportovať len jednu kategóriu metrík.

Používali sa 2 formáty výstupu podľa spôsobu zaobchádzania s vektorovými atribútmi. Na základe konfiguračného parametru VECTORS_AS_LIST sa generujú jednotlivé položky atribútu buď ako samostatný atribút alebo ako jeden vektor. Vzhľadom na to, že generovanie výstupu nie je časovo náročné, sa pristúpilo k variante kedy sa vygenerujú oba formáty súčasne počas jedného behu extrakcie metrík. Ukážky výstupov v oboch formátoch sú na obrázkoch č. 5.3 a 5.4.

id class label	distributed	distributed	distributed	distributed	distributed	distributed
id label	lnPkt1s10i[0]	lnPkt1s10i[1]	lnPkt1s10i[2]	lnPkt1s10i[3]	lnPkt1s10i[4]	lnPkt1s10i[5]
1 False	0	1	0	1	2	1
2 False	1	3	0	1	0	3
3 False	1	0	3	1	0	3
4 False	1	3	0	1	3	0
5 False	1	3	0	1	2	1
6 False	1	2	1	0	1	2
7 False	1	2	1	0	1	3
8 False	3	1	0	1	0	4
9 False	1	3	1	1	3	0
10 False	3	1	0	1	2	1
11 False	1	3	0	0	1	2
12 False	0	1	2	1	0	1
13 False	3	1	0	1	2	2
14 False	1	3	1	1	1	3

Obrázok 5.3: Ukážka výstupu so separovanými položkami vektorov.

id class label	distributed	distributed	distributed
id label	lnPkt1s10i	lnPkt4s10i	lnPkt8s10i
1 False	[0, 1, 0, 1, 2, 1, 0, 3, 1, 0]	[2, 6, 4, 5, 4, 4, 4, 4, 8, 4]	[8, 9, 8, 8, 12, 8, 10, 9, 8, 8]
2 False	[1, 3, 0, 1, 0, 3, 0, 1, 3, 0]	[5, 4, 7, 3, 5, 4, 5, 4, 4, 4]	[9, 10, 9, 9, 8, 10, 10, 7, 9, 8]
3 False	[1, 0, 3, 1, 0, 3, 0, 1, 3, 0]	[5, 4, 6, 4, 5, 2, 7, 4, 4, 4]	[9, 10, 7, 11, 8, 10, 9, 8, 9, 13]
4 False	[1, 3, 0, 1, 3, 0, 1, 2, 0, 1]	[5, 6, 4, 4, 2, 8, 4, 4, 4, 5]	[11, 8, 10, 8, 9, 10, 8, 8, 9, 14]
5 False	[1, 3, 0, 1, 2, 1, 0, 1, 2, 1]	[5, 4, 4, 4, 5, 6, 2, 6, 3, 8]	[9, 8, 11, 8, 11, 8, 9, 8, 14, 8]
6 False	[1, 2, 1, 0, 1, 2, 1, 0, 1, 3]	[4, 4, 4, 5, 5, 4, 4, 4, 6, 7]	[8, 9, 9, 8, 13, 8, 9, 7, 15, 8]
7 False	[1, 2, 1, 0, 1, 3, 0, 0, 1, 3]	[4, 4, 5, 5, 4, 4, 4, 6, 7, 2]	[8, 10, 8, 10, 9, 10, 9, 13, 9, 8]
8 False	[3, 1, 0, 1, 0, 4, 0, 1, 3, 0]	[5, 5, 4, 4, 4, 5, 8, 5, 1, 7]	[10, 8, 9, 13, 8, 8, 8, 14, 8, 8]
9 False	[1, 3, 1, 1, 3, 0, 0, 1, 3, 0]	[6, 4, 4, 4, 4, 8, 6, 1, 6, 4]	[10, 8, 12, 7, 10, 9, 13, 9, 8, 12]
10 False	[3, 1, 0, 1, 2, 1, 0, 1, 0, 3]	[5, 4, 4, 4, 6, 7, 2, 6, 4, 5]	[9, 8, 13, 8, 9, 7, 15, 8, 8, 15]
11 False	[1, 3, 0, 0, 1, 2, 1, 0, 1, 3]	[4, 4, 5, 5, 5, 5, 4, 4, 4, 6]	[8, 10, 10, 8, 10, 8, 15, 8, 8, 18]
12 False	[0, 1, 2, 1, 0, 1, 3, 0, 1, 0]	[4, 4, 5, 5, 6, 3, 4, 5, 4, 4]	[8, 10, 9, 9, 8, 14, 9, 7, 12, 15]
13 False	[3, 1, 0, 1, 2, 2, 1, 0, 4, 0]	[5, 5, 7, 3, 4, 4, 7, 4, 3, 8]	[10, 10, 8, 11, 11, 11, 8, 11, 16,
14 False	[1, 3, 1, 1, 1, 3, 0, 1, 3, 1]	[6, 5, 5, 4, 3, 5, 4, 4, 5, 9]	[11, 9, 8, 8, 14, 11, 6, 11, 17, 5]

Obrázok 5.4: Ukážka výstupu s vektormi ako samostatnými atribútmi.

Kapitola 6

Analýza v nástroji RapidMiner

V rámci tejto kapitoly bude uskutočňovaná analýza dátovej kolekcie KDD Cup 1999 a následná analýza vyexportovaného výstupného DSV súboru, ktorý obsahuje extrahované metriky jednotlivých spojení. Najskôr sa bude robiť analýza jednotlivých metrík s členením vektorových reprezentácií na samostatné prvky. Bude sa tu skúmať kvalita a relevantnosť jednotlivých metrík.

Potom sa bude experimentovať s viacerými metrikami na vytvorenie vhodného klasifikačného modelu. Vstupný súbor vytvorenej dátovej kolekcie metrík obsahuje spolu 185 spojení, z čoho je 12 útokov. Verzia nástroja RapidMiner, ktorá bola v rámci tejto práce použitá je 5.2.003. Uvedené blokové schémy zapojenia budú použité z tejto verzie nástroja RapidMiner.

6.1 Import vstupu

Pri samotnom importovaní dát v nástroji RapidMiner je potrebné uskutočniť niekoľko potrebných krokov smerujúcich k správnej interpretácii vstupných dát. Ako prvé je nutné zvoliť správny oddeľovač jednotlivých polí vo vstupnom súbore. Oddeľovačov môže byť súčasne použitých aj viacej a je možné ich špecifikovať regulárnym výrazom. V ďalšom kroku nasleduje anotácia jednotlivých atribútov, kde sa určí, ktoré riadky sú komentármi, ktorý riadok obsahuje názvy jednotlivých atribútov a kde začínajú samotné dátové vektory. Po tomto kroku je nutné nastaviť správny typ a rolu každého atribútu. Rola atribútu v nástroji RapidMiner určuje význam atribútu. Role, ktoré boli nastavené počas importu použitého vstupu sú:

- rola **identifikácie** záznamu – označená ako *id*,
- rola **návestia** záznamu – udáva triedu, do ktorej záznam patrí a je označená ako *label*,
- rola **atribútu** záznamu – generická rola špecifikujúca štandardný atribút. Je označená ako *attribute*.

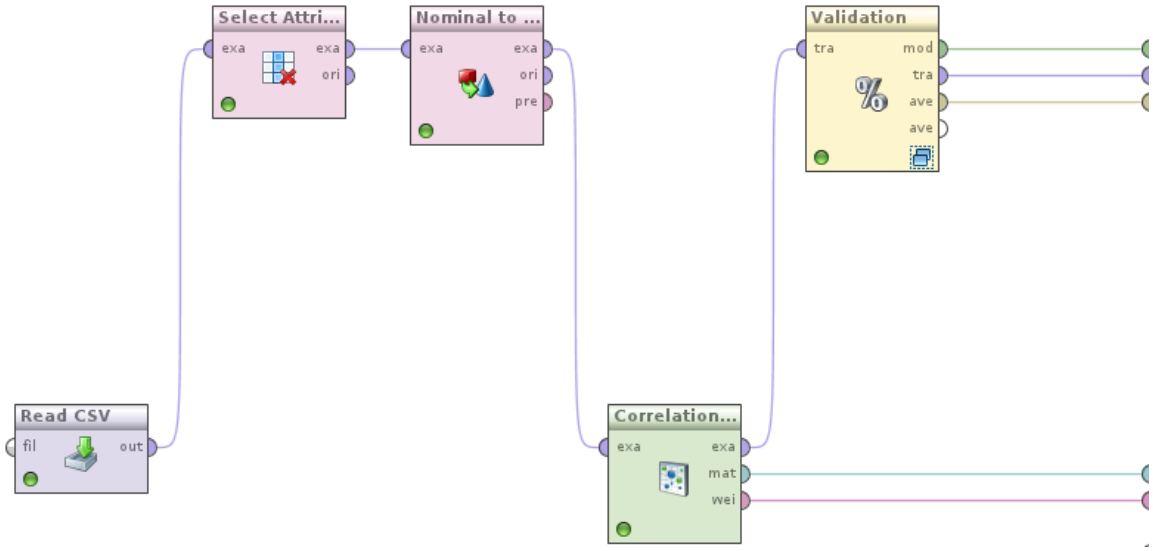
Na koniec je potrebné nastaviť typ každého atribútu. Nástroj RapidMiner používa implicitne odhadovanie typov podľa vzorky vstupných dát. V každom prípade sa na tento odhad nedá spoliehať a typy treba prekontrolovať a nastaviť správne. Jedná sa o nominálne atribúty, ktoré sú odhadnuté ako atribúty typu integer, ktorými sú zdrojový a cieľový port.

Ďalej sa jedná o nominálne atribúty zdrojová MAC adresa, cieľová MAC adresa, zdrojová IP adresa a cieľová IP adresa, ktoré sú odhadnuté ako binominálne. Príčinou tohto nesprávneho odhadu sú laboratórne podmienky, v ktorých boli experimenty realizované a dochádzalo ku komunikácii len medzi dvomi IP adresami a tiež MAC adresami.

6.2 Bloková schéma procesu dátovej analýzy

Bloková schéma procesu nástroja RapidMiner, ktorý bol použitý na analýzu metrík je znázornená na obrázku č. 6.1. Prvý blok *Read CSV* uskutočňuje načítanie vstupu z CSV súboru, kde jednotlivým atribútom pridelí typy a role. Ďalší blok *Select Attribute* vyberá podmnožinu zo všetkých atribútov. Je to z dôvodu, že niektoré atribúty môžu byť korelované alebo zbytočné. Blok *Nominal to Numerical* je voliteľný a slúži na konverziu nominálných atribútov na numerické špeciálnym spôsobom popísanom v podsekcii č. 6.2.1. Nasleduje blok *Correlation matrix*, ktorý zo vstupu vygeneruje korelačnú maticu a vstup posiela do nasledujúceho bloku nemodifikovaný. Na konci blokovej schémy je blok *Validation* slúžiaci na vyhodnotenie úspešnosti klasifikácie použitého operátoru učenia (resp. jeho modelu). Jeho podrobný popis aj s internou schémou zapojenia je uvedený v podsekcii č. 6.2.2. Výstupy bloku *Validation* sú:

- *mod* – reprezentácia modelu použitej klasifikačnej metódy. V prípade naivnej Bayesovej metódy je to rozloženie hustoty jednotlivých atribútov, v prípade rozhodovacieho stromu je to grafická a textová reprezentácia stromu, v prípade neurónovej siete je to grafická schéma neurónovej siete aj z jednotlivými váhami, v prípade SVM je to zoznam váh pre jednotlivé atribúty a tabuľka *support* vektorov.
- *tra* – štatistiky všetkých vstupných atribútov vrátane strednej hodnoty, smerodajnej odchýlky a rozsahu hodnôt,
- *ave* – performance vector udávajúci celkovú kvalitu klasifikácie a matica zámen (confusion matrix) udávajúca percentuálne podiely klasifikácie jednotlivých vzoriek do cieľových tried [1].



Obrázok 6.1: Bloková schéma procesu analýzy metrík.

6.2.1 Konverzia nominálných atribútov

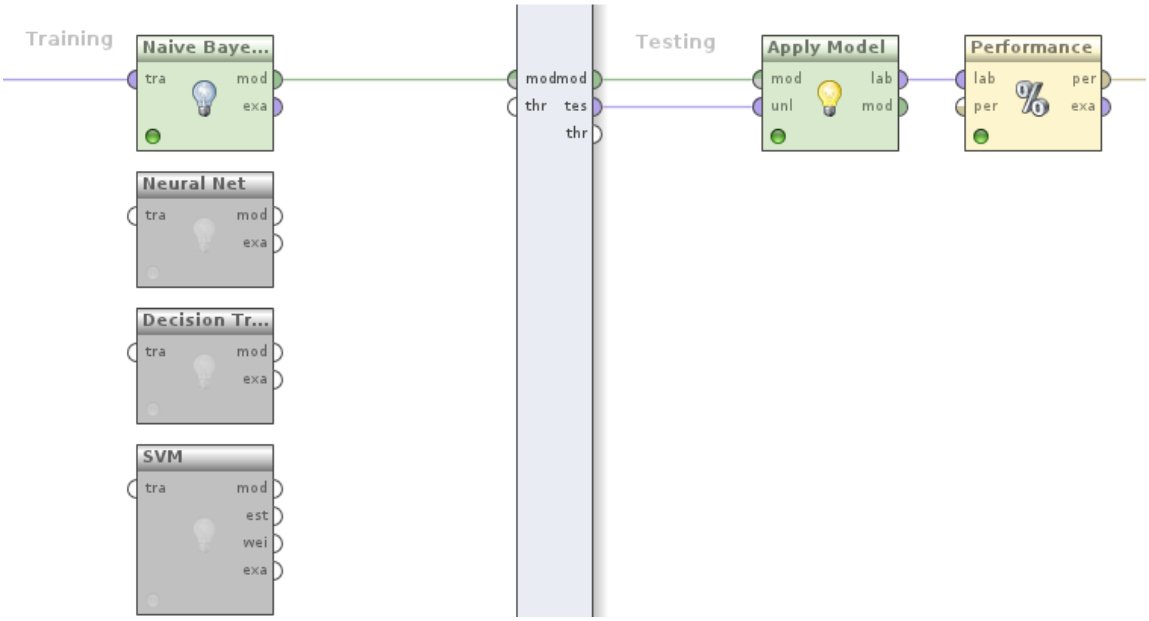
Niektoré klasifikačné modely použité pri validácii vyžadujú určité typy atribútov. Preto je niekedy potrebné prekonvertovať nominálne atribúty do inej reprezentácie. Nástroj RapidMiner uskutočňuje konverziu nominálnych atribútov na numerické jedným z dvoch spôsobov:

- **dichotomizáciou** – je definovaná komparačná skupina, pre ktorú je zvolená jedna hodnota nominálneho atribútu. Potom je pre všetky hodnoty nominálnych atribútov (okrem komparačnej skupiny) vytvorený nový atribút. Potom nový atribút, ktorý korešponduje z aktuálnou nominálnou hodnotou má hodnotu 1 a všetky ostatné atribúty majú hodnotu 0. Keď hodnota nominálneho atribútu korešponduje komparačnej skupine, tak všetky nové atribúty sú nastavené na 0,

- **rovnocenné ohodnotenie** – hodnota nominálneho atribútu je braná rovnocenne ohodnotená, a preto jeho hodnota bude jednoducho prevedená na reálny atribút. Vzdialenosti medzi jednotlivými hodnotami sú ekvidištančné [3]. Jedná sa implicitnú metódu nástroja RapidMiner pre konverziu nominálnych atribútov na numerické. Preto v rámci experimentov bola využívaná táto metóda.

6.2.2 Validácia modelov

Ako prostriedok validácie schopný vyhodnotiť odhad výkonnosti jednotlivých modelov je použitá krížová validácia, ktorú v nástroji RapidMiner poskytuje blok *Validation*. Vstupná sada vzoriek je rozdelená na n skupín S_i, kde n reprezentuje počet validácií a S_i i-tu skupinu. Vnútorný subproces tohto bloku aplikuje n-krát proces trénovania a testovania. Ako testovacia sada sa v každom kroku i použije S_i a ako trénovacia sada sa použije $S \setminus S_i$. Výsledkom trénovacieho subprocesu je model natrénovaný na trénovacích dátach. Testovací subproces vracia *Performance* vektor. Toto je obyčajne dosiahnuté aplikovaním modelu zmeraním jeho výkonu [4]. Bloková schéma vnútorných procesov použitého validačného bloku je znázornená na obrázku č. 6.2. V ľavej časti schémy je možné vidieť sadu trénovacích modelov, z ktorej je stále aktívny len jeden (v uvedenom príklade je to naivný Bayessovský). Ostatné používané modely zobrazené šedo, sú neaktívne. Výstup trénovacieho procesu je model (*mod*), ktorý je privedený ako vstup testovaciemu procesu. Ďalším vstupom sú testovacie dáta (*tes*). Testovací proces obsahuje komponent *Apply model*, ktorý použije natrénovaný model na klasifikáciu nových vzoriek z testovacej sady. Nasledujúci blok *Performace,* uskutoční ohodnotenie kvality klasifikácie testovacích dát a svoj výstup posiela na výstup komponentu *Validation,* ktorý z týchto výsledkov vypočíta *Performance vector* a tiež maticu zámen.



Obrázok 6.2: Bloková schéma validačného bloku.

6.3 Analýza KDD Cup 1999

V tejto časti práce sa bude hľadať vhodný model, ktorý dokáže komunikácie kolekcie KDD Cup 1999 správne klasifikovať. Bude tu použitá krížová validácia pre zistenie kvality klasifikačného modelu.

Najlepší výsledok, ktorý sa podarilo dosiahnuť bol uskutočnený naivnou Bayessovou metódou a je znázornený formou matice zámen v tabuľke č. 6.1. Je možné vidieť, že k správnej detekcii útoku došlo v 95.77% prípadoch a legitímne komunikácie boli nesprávne klasifikované len v 0.77% prípadoch. (Celková úspešnosť klasifikácie je 97.55% +/- 0.08%).

	true False	**true True**	**presnosť tried**
predikované False	13345	497	96.41%
predikované True	104	11246	99.08%
odozva modelu	99.23%	95.77%	

Tabuľka 6.1: Matica zámen pre naivnú Bayessovskú metódu (kolekcia KDD Cup 1999).

Ostatnými metódami neboli nájdené lepšie klasifikačné modely. Za prvé to nebolo cieľom tejto práce a za druhé s touto dátovou kolekciou už bolo v minulosti dostatočne experimentované a výsledky boli diskutované v publikáciách [29].

6.4 Analýza rozloženia hustoty navrhnutých metrík

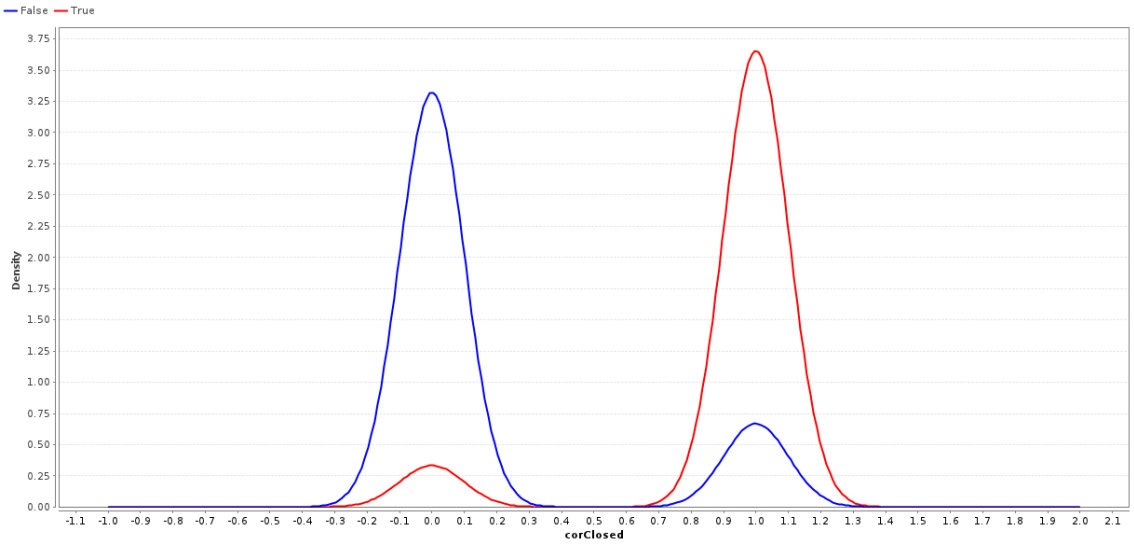
Navrhnuté a implementované metriky boli analyzované pomocou schémy uvedenej v sekcii č. 6.2. Bol tu použitý naivný Bayessovský klasifikátor [2] a vizuálne sa analyzovalo rozloženie hustoty každého atribútu vzhľadom na príslušnosť do danej triedy. Bolo identifikovaných 45 metrík, ktoré vykazovali zaujímavé rozloženie hustoty prvkov vstupnej kolekcie. V každom grafe, ktorý bude reprezentovať hustotu rozloženia danej metriky, bude príslušnosť do

tried znázornená červenou farbou pre triedu reprezentujúcu útoky (True) a modrou farbou pre triedu reprezentujúcu legitímne komunikácie (False). Jedná sa o metriky:

- corClosed – korektné ukončenie spojenia. Rozloženie hustoty metriky je znázornené v grafe na obrázku č. 6.3,
- distribučné metriky – InPkt32s10i[2], InPkt32s10i[5], InPkt64s10i[1], InPkt64s20iTr1KB[0], InPkt64s20iTr1KB[1], OutPkt1s10i[1], OutPkt1s20iTr4KB[4], OutPkt32s10i[2], OutPkt64s20iTr1KB[1], OutPkt64s20iTr1KB[2], OutPkt64s20iTr2KB[1], OutPktLen4s10i[0], OutPktLen4s10i[2], OutPktLen8s10i[3], z ktorých ako reprezentatívny príklad je vybratá metrika OutPkt64s20iTr1KB[1], ktorej hustota rozloženia je zobrazená v grafe na obrázku č. 6.4,
- štatistické metriky meanPktLenSrc, sigPktLenSrc, sumPktLenSrc, ratInOutPkt, cntNondPktIn – priemerná hodnota dĺžky zdrojových paketov, smerodajná odchýlka dĺžky zdrojových paketov, suma dĺžok zdrojových paketov, pomer zdrojových ku cieľovým paketom a počet nedátových cieľových paketov. Rozloženie hustoty reprezentatívnej metriky tejto skupiny meanPktLenSrc je znázornené v grafe na obrázku č. 6.5. Ostatné rozloženia hustoty uvedených metrík sa nachádzajú v prílohe A.1.
- dynamické metriky cntResendPktsIn, SigTdiff2PktsIn – počet znova poslaných cieľových paketov a smerodajná odchýlka časov medzi dvomi paketmi v smere dnu. Rozloženie hustoty reprezentatívnej metriky SigTdiff2PktsIn je znázornené v grafe na obrázku č. 6.6. Rozloženie hustoty metriky cntResendPktsIn je uvedené v prílohe A.2.
- metriky, ktoré aproximujú priebehy komunikácií polynómami – polynomIndexes3ordOut[0], polynomIndexes3ordOut[2], polynomIndexes3ordOut[3], polynomIndexes8ordIn[7], z ktorých ako reprezentatívny príklad je vybratá metrika polynomIndexes3ordOut[3], ktorej hustota rozloženia je zobrazená v grafe na obrázku č. 6.7,
- metriky, ktoré aproximujú komunikácie Fourierovými radami – fourCoefsGonAngleOut[14], fourCoefsGonAngleOut[1], fourCoefsGonAngleOut[3], fourCoefsGonAngleOut[8], fourCoefsGonModulOut[0], fourCoefsGonModulOut[1], fourCoefsGonModulOut[2], fourCoefsGonModulOut[4], fourCoefsGonAngleIn[7], fourCoefsGonmodulIn[3], fourCoefsGonModulIn[5], z ktorých ako reprezentatívny príklad je vybratá metrika fourCoefsGonAngleOut[3], ktorej hustota rozloženia je zobrazená v grafe na obrázku č. 6.8,
- normalizované produkty komunikácie s gaussovými krivkami – gaussProds2Out[1], gaussProds4In[0], gaussProds4Out[0], gaussProds4Out[2], gaussProds8All[7], gaussProds8Out[2], gaussProds8Out[3], gaussProds8Out[7], z ktorých ako reprezentatívny príklad je vybratá metrika gaussProds2Out[1], ktorej hustota rozloženia je zobrazená v grafe na obrázku č. 6.9.

6.5 Naivná Bayessovská metóda

Na základe identifikácie metrík s vhodným rozložením hustoty sa pristúpilo ku selekcii podmnožiny metrík a s týmito sa ďalej experimentovalo. Bolo tak učinené z dôvodu aby algoritmy učenia boli schopné rýchleho behu aj vo výpočtovo náročnejších konfiguráciách.



Obrázok 6.3: Rozloženie hustoty metriky corClosed.

Ďalším intuitívnym dôvodom je poskytnúť do procesu klasifikácie relevantné atribúty a sústrediť na ne činnosť klasifikačného elementu.

V tomto experimente sa pracovalo s modelom naivnej Bayessovskej metódy klasifikácie, ktorá sa využívala na vyhodnotenie kvality klasifikácie na celej dátovej kolekcii obsahujúcej extrahované metriky. Pre určenie kvality odozvy natrénovaného modelu bola využitá krížová validácia. V tomto experimente bolo využité stratifikované vzorkovanie [6], ktoré používa rovnomerné zastúpenia jednotlivých tried pre trénovanie modelu. Výsledok kvality odozvy modelu určuje matica zámen, ktorá je pre tento experiment uvedená v tabuľke č. 6.2. Celková úspešnosť klasifikácie je *87.57%*.

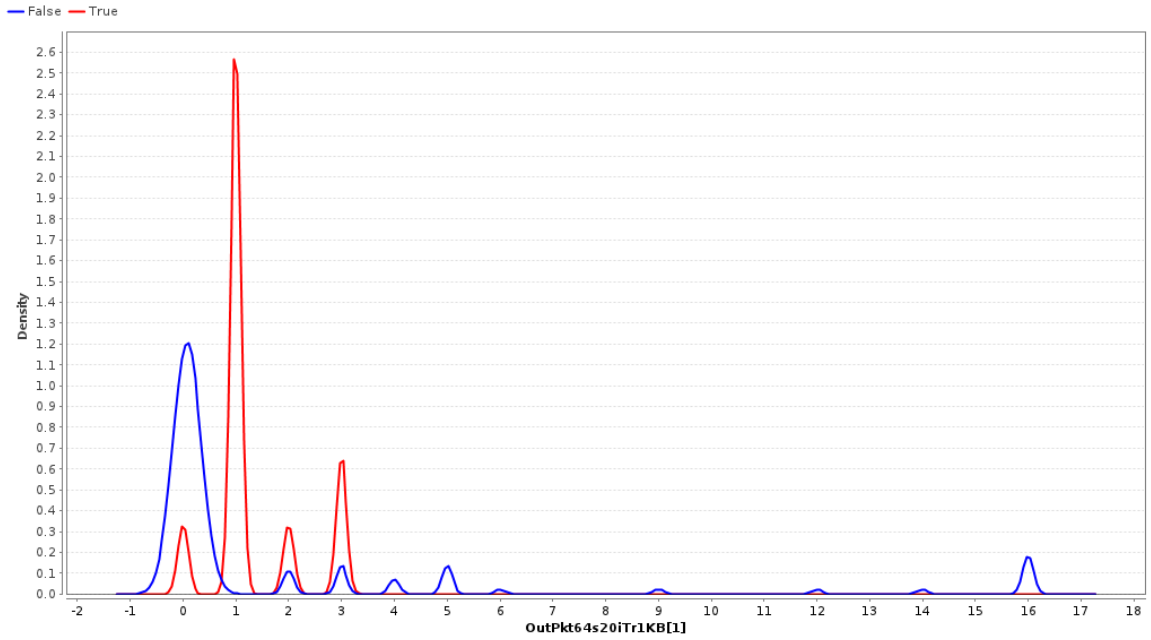
	true False	**true True**	**presnosť tried**
predikované False	151	1	99.34%
predikované True	22	11	33.33%
odozva modelu	87.28%	91.67%	

Tabuľka 6.2: Matica zámen pre naivnú Bayessovskú metódu.

K detekcii útoku došlo došlo v 91.76% prípadoch útokov, čo je na použitý model dobrý výsledok. K detekcii legitímnej komunikácie došlo len v 87.28% a teda 12.72% legitímnych komunikácií bolo prehlásených za útok.

6.6 Metóda PCA

Na základe analýzy korelačnej matice, ktorá je jedným z výstupov základnej blokovej schémy zapojenia komponentov, bola zistená korelácia niektorých atribútov. Preto bolo učinené rozhodnutie vyskúšať metódu extrakcie hlavných komponentov PCA [21], ktorá redukuje korelované atribúty. V tomto experimente sa pracovalo s modelom naivnej Bayessovskej metódy klasifikácie. S implicitnými nastaveniami, ktoré zahrňovali automatické stanovenie počtu hlavných komponent neboli ani zďaleka dosahované výsledky ako tie, ktoré boli dosiahnuté naivným Bayessovským klasifikátorom bez PCA. Počet komponentov, ktorý bol



Obrázok 6.4: Rozloženie hustoty metriky OutPkt64s20iTr1KB[1].

metódou určený bol 3. Matica zámen pre tento experiment je situovaná v tabuľke č. 6.3. Celková úspešnosť klasifikácie je *91.92%*.

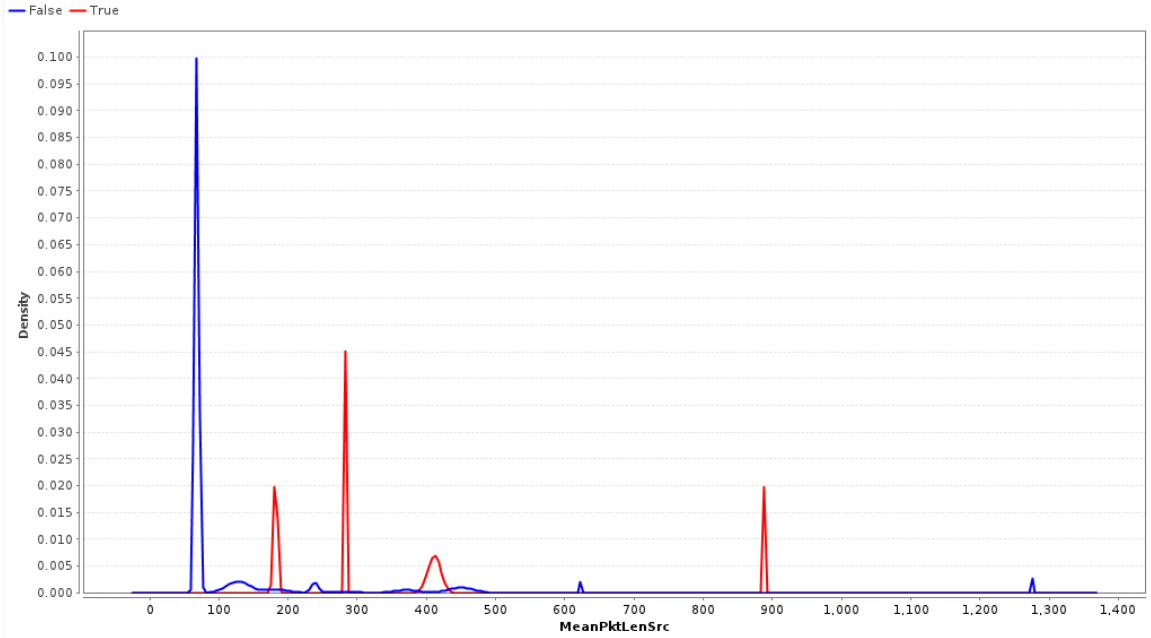
	skutočné False	**skutočné True**	**presnosť tried**
predikované False	168	10	94.38%
predikované True	5	2	28.57%
odozva modelu	97.11%	16.16%	

Tabuľka 6.3: Matica zámen pre PCA s určením počtu komponentov.

Nasledovne boli upravené nastavenia tohto komponentu, kde sa experimentovalo s fixne stanoveným počtom hlavných komponentov, ktoré sa majú z množiny atribútov extrahovať. V porovnaní s implicitnými nastaveniami sa prinesené výsledky o málo zlepšili oproti predchádzajúcemu príkladu, no opäť sa nepribližovali výsledkom ako vo variante bez tohto komponentu. Matica zámen pre tento experiment je zobrazená v tabuľke č. 6.4. Celková úspešnosť klasifikácie je *88.10%*.

	skutočné False	**skutočné True**	**presnosť tried**
predikované False	153	2	98.71%
predikované True	20	10	33.33%
odozva modelu	88.44%	83.33%	

Tabuľka 6.4: Matica zámen pre PCA s manuálnym nastavením počtu komponentov.



Obrázok 6.5: Rozloženie hustoty metriky meanPktLenSrc.

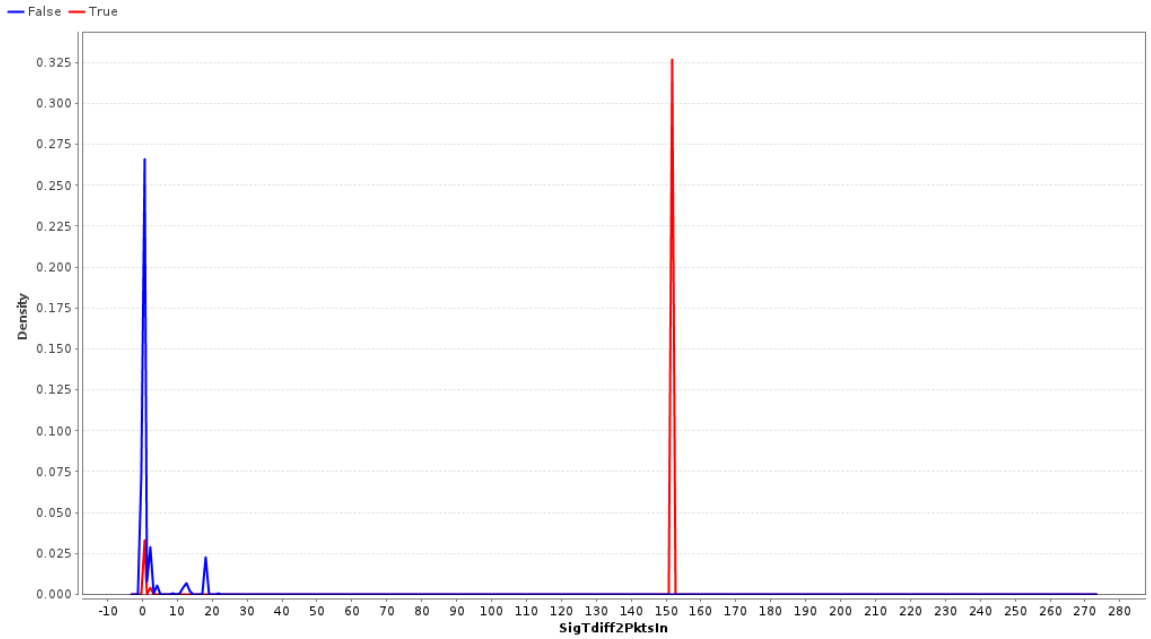
6.7 Zavedenie diskretizácie ordinárnych atribútov

Vzhľadom na charakter rozloženia hustoty jednotlivých atribútov, ktoré boli analyzované v sekcii č. 6.4, bolo pristúpené diskretizáciu použitých atribútov binningom, ktoré uskutočňuje diskretizáciu všetkých numerických ordinárnych atribútov na nominálne. V tomto experimente sa pracovalo s modelom naivnej Bayessovskej metódy klasifikácie. Rozsahy numerických atribútov sú rozdelené na segmenty rovnakej veľkosti. Každý segment reprezentuje kôš, do ktorého sú pridelené jednotlivé položky podľa číselnej hodnoty daného atribútu [5]. Pre uvedený výsledok bolo použitých 5 košov v prípade každej metriky. Matica zámen pre tento experiment je uvedená v tabuľke č. 6.5. Celková úspešnosť klasifikácie je *94.06%*.

	skutočné False	**skutočné True**	**presnosť tried**
predikované False	163	1	99.39%
predikované True	10	11	52.38%
odozva modelu	94.22%	91.67%	

Tabuľka 6.5: Matica zámen pre diskretizáciu ordinárnych atribútov.

V tomto prístupe bola dosiahnutá rovnako dobrá úspešnosť klasifikácie útokov ako v prípade bez diskretizácie, ale prínos tohto prístupu spočíva v lepšej klasifikácii legitímnych komunikácií. Keďže v prípade naivnej Bayessovej metódy bez diskretizácie ordinárnych atribútov bolo nesprávne klasifikovaných 12.72% legitímnych komunikácií, tak v tomto prípade to bolo len 5.78%, čo je o vyše polovicu menej.



Obrázok 6.6: Rozloženie hustoty metriky SigTdiff2PktsIn.

6.8 SVM

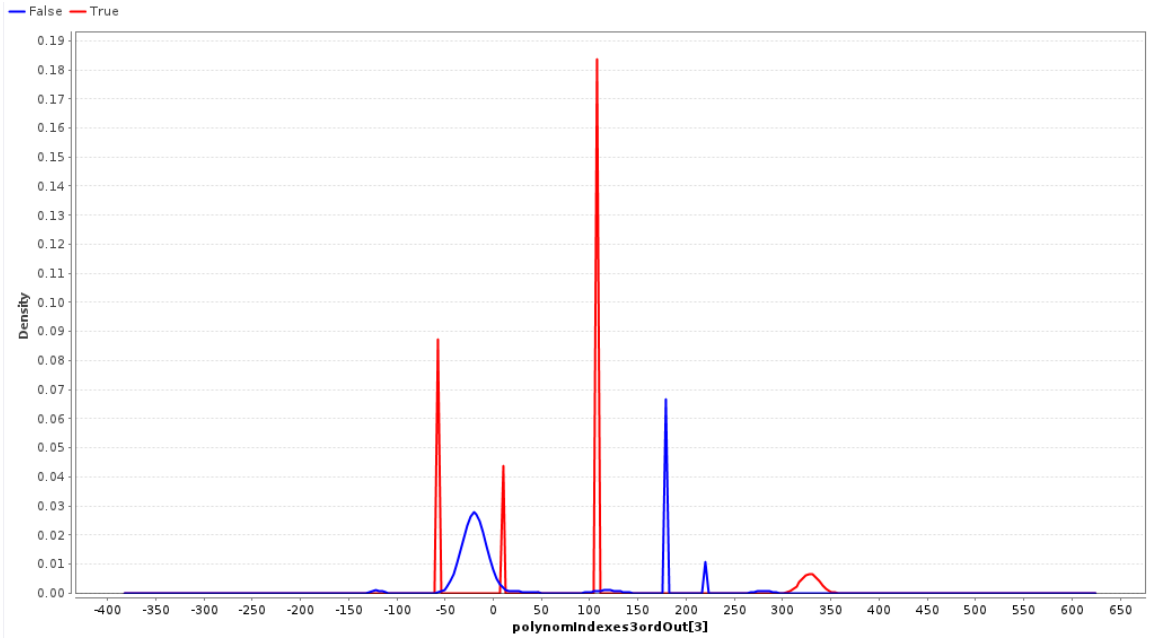
V tomto experimente sa tiež experimentovalo len s podmnožinou metrík, ktoré boli identifikované ako významné v sekcii č. 6.4. Diskretizácia tu použitá nebola, keďže SVM vyžaduje prácu s numerickými atribútmi. Experimentovalo sa s typom kernelu. Ako najúspešnejší sa ukázal neurónový. Rovnako ako v predchádzajúcich experimentoch, tak aj to bolo využité stratifikované vzorkovanie a krížová validácia. Taktiež sa experimentovalo z cenovým parametrom, ktorý určuje akceptovateľnú presnosť klasifikácie do jednotlivých tried. V našom prípade určuje hranicu medzi legitímnymi komunikáciami a útokmi. Inak povedané, určuje do akej miery sme ochotní klasifikovať legitímnu komunikáciu ako útok vzhľadom na to, aby čo najviac útokov bolo klasifikovaných ako útok. Jedná sa o posúvanie kritéria ROC krivky[1]. Pri iných cenových parametroch (iných nastaveniach ROC kritéria) sa nepodarilo nájsť optimálnejšie konfigurácie ako doposiaľ nájdené. Preto bol tento parameter zvolený tak aby detekoval všetky útoky s určitými nepresnosťami detekcie legitímnych komunikácií. Najlepšie výsledky dosiahnuté experimentami s týmto modelom sú znázornené vo forme matice zámen v tabuľke č. 6.6. Celková úspešnosť klasifikácie je *82.17%*.

	skutočné False	**skutočné True**	**presnosť tried**
predikované False	140	0	100.00%
predikované True	33	12	26.67%
odozva modelu	80.92%	100.00%	

Tabuľka 6.6: Matica zámen pre model SVM.

V rámci tohto experimentu bola dosiahnutá 100%-ná úspešnosť klasifikácie útokov. Problémom zostáva vysoký počet legitímnych komunikácií, klasifikovaných ako útoky (19.08%).

[1]Receiver operating characteristic. http://gim.unmc.edu/dxtests/ROC1.htm.



Obrázok 6.7: Rozloženie hustoty metriky polynomIndexes3ordOut[3].

6.9 Rozhodovací strom

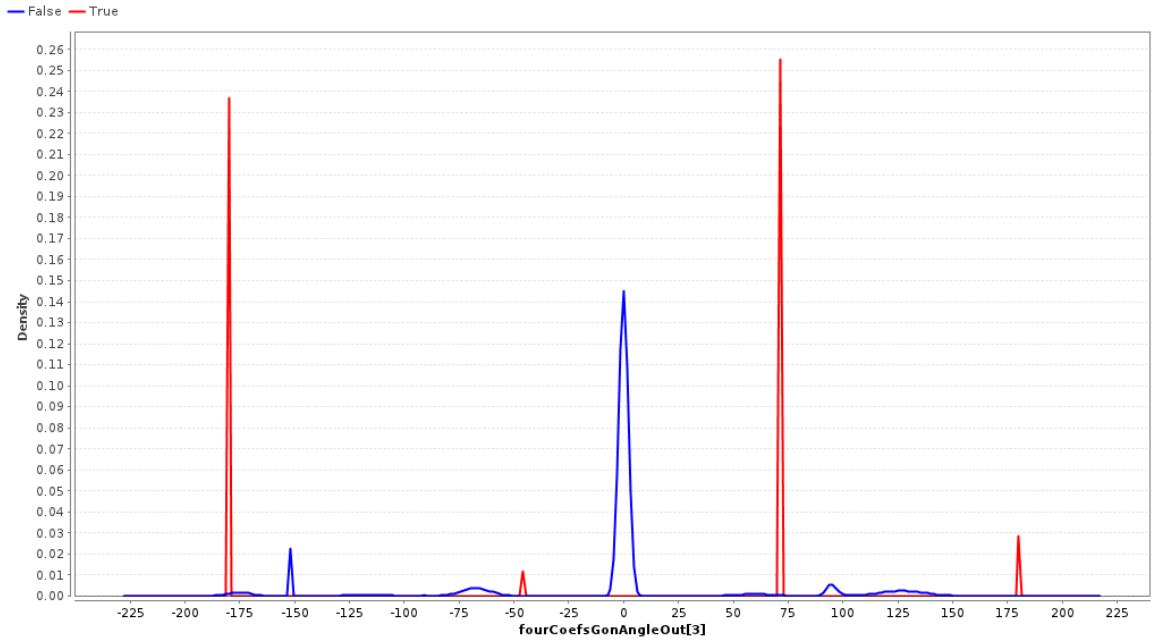
Ako ďalší model klasifikácie vstupnej sady dát bol použitý rozhodovací strom. Pre určenie kvality odozvy natrénovaného modelu bola opäť využitá krížová validácia. V tomto experimente bolo využité stratifikované vzorkovanie [6], ktoré používa rovnomerné zastúpenia jednotlivých tried pre trénovanie modelu. Bolo tu experimentované s typom kritéria rozhodovacieho stromu, kde sa ukázala ako najvhodnejšia voľba *gini index*[2] kritéria. Výsledok kvality odozvy modelu určuje matica zámen, ktorá je pre tento experiment uvedená v tabuľke č. 6.7. Celková úspešnosť klasifikácie je *95.14%*.

	skutočné False	**skutočné True**	**presnosť tried**
predikované False	169	5	97.13%
predikované True	4	7	63.64%
odozva modelu	97.69%	58.33%	

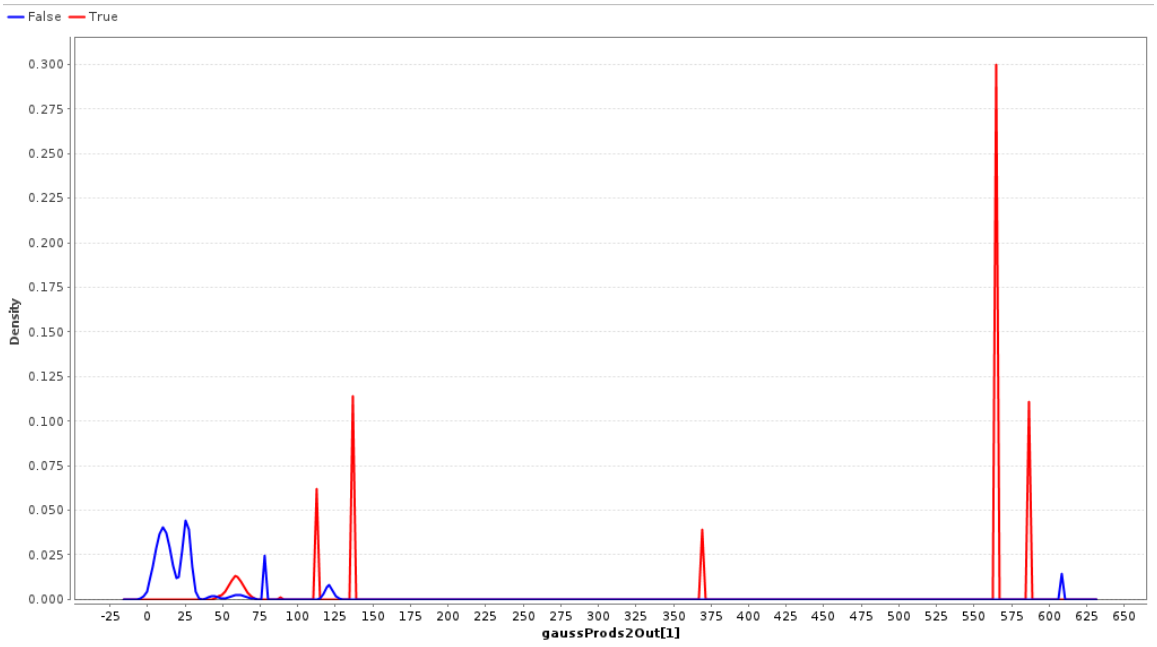
Tabuľka 6.7: Matica zámen pre model rozhodovacieho stromu.

V tomto experimente sa podarilo nájsť model klasifikačného stromu, ktorý umiestňuje kritérium ROC krivky do miesta, kde je možné detekovať maximálny počet legitímnych komunikácií, zatiaľ čo v ostatných experimentoch bolo toto kritérium volené tak, aby bol detekovaný maximálny počet útokov. Správne klasifikovaných legitímnych komunikácií bolo 97.69%, pričom bolo detekovaných 58.33% útokov. Grafická reprezentácia nájdeného rozhodovacieho stromu generovaná nástrojom RapidMiner je uvedená v prílohe B.1.

[2]Kritérium voľby atribútu rozhodovacieho stromu. http://www.hypertextbookshop.com/-dataminingbook/working_version/contents/chapters/chapter001/section003/blue/page002.html.



Obrázok 6.8: Rozloženie hustoty metriky fourCoefsGonAngleOut[3].



Obrázok 6.9: Rozloženie hustoty metriky gaussProds2Out[1].

6.10 Zhrnutie výsledkov

Sumarizácia výsledkov pre jednotlivé klasifikačné metódy, ktoré boli použité v tejto práci je zhrnutá v tabuľke č. 6.8. Je možné si všimnúť, že jediná metóda, ktorá bola schopná správne klasifikovať všetky útoky je SVM. SVM malo na druhej strane značný problém s klasifikáciou legitímnych komunikácií, čo nebolo možné dostatočne zmierňovať ani experimentovaním s cenovým parametrom. Legitímne komunikácie boli najlepšie klasifikované pri použití rozhodovacieho stromu, no bolo tu detekovaných len o niečo viac ako polovica útokov. Výsledok tejto metódy je možné efektívne uplatniť v prostrediach, kde je výpadok systému drahší ako samotný útok. Dostupnosť systému je maximalizovaná a stále je detekovaná dosť značná časť útokov.

Iný dosiahnutý pokrok, ktorý je kompromisom spomenutých dvoch výsledkov je naivná Bayessova metóda s diskretizáciou atribútov. V tejto metóde je správne klasifikovaných vyše 90% legitímnych komunikácií a aj útokov.

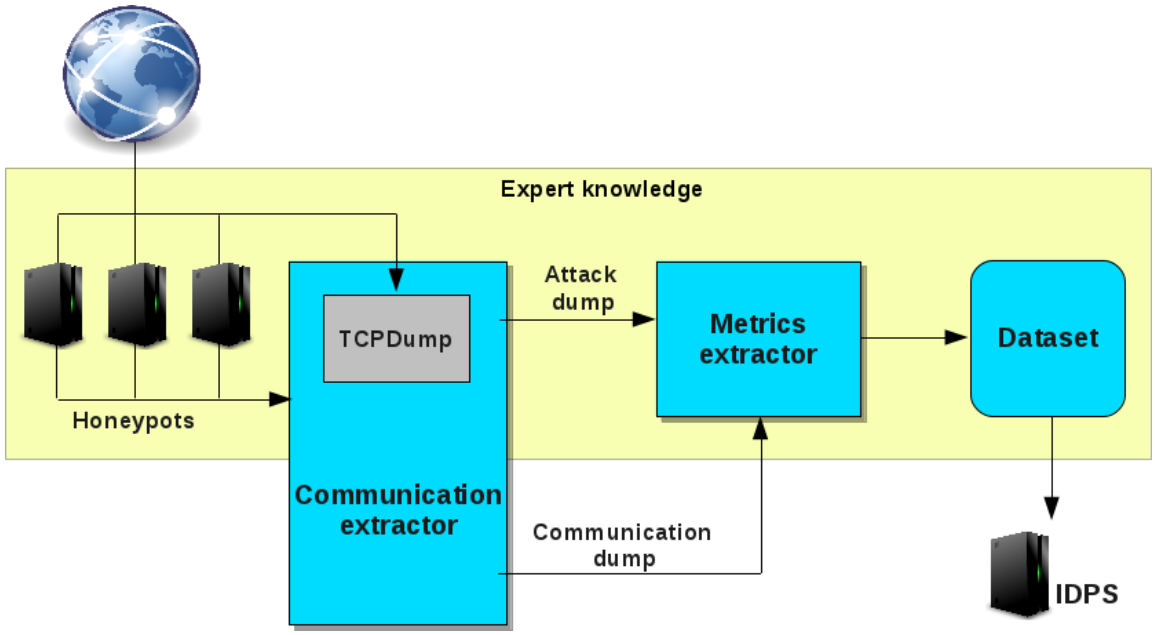
Klasifikačná metóda	**Odozva detekcie legitímnych komunikácií**	**Odozva detekcie útokov**	**Celková úspešnosť klasifikácie**
SVM s neurónovým kernelom	80.92%	100.00%	82.17%
Naivná Bayesova metóda	87.28%	91.67%	87.57%
Naivná Bayesova metóda (PCA pri fixnom počte komponentov)	88.44%	83.33%	88.10%
Naivná Bayesova metóda (PCA pri automatickom počte komponentov)	97.11%	16.16%	91.92%
Naivná Bayesova metóda s diskretizáciou atribútov	94.22%	91.67%	94.06%
Rozhodovací strom (kritérium selekcie atribútov – gini index)	97.69%	58.33%	95.14%

Tabuľka 6.8: Zhrnutie výsledkov klasifikácie jednotlivých modelov.

Kapitola 7

Návrh modelu architektúry nasadenia

Model architektúry, ktorý je braný počas tejto práce v úvahu je znázornený na obrázku č. 7.1. V tomto modely sú brané v úvahu expertné znalosti, ktoré presne definujú ako vyzerá správanie útoku. Pre tento účel je použitý *shadow honeypot* [14] systém pre detekciu nových útokov. Vzhľadom na princíp činnosti použitého honeypotu Argos sú skúmané len *buffer overflow* útoky. V modely je využitý TCPdump počúvajúci na sieťových rozhraniach honeypotov. Ďalšími časťami modelu sú extraktor komunikácií (Communication extractor) a extraktor metrík (Metrics extractor), ktoré pracujú nad dátami získanými TCPdumpom. Tieto dve časti sú použité na extrakciu komunikácií a následne metrík z nazbieraných dát. Ďalšou časťou modelu je komponent IDPS, ktorý využíva množinu metrík extrahovaných predchádzajúcimi časťami k detekcii a predchádzaniu útokov u ostatných strojoch korporátnej siete. Na obrázku č. 7.1 sú znázornené tri honeypoty s rôznymi operačnými systémami a rôznymi zraniteľnými službami [17].



Obrázok 7.1: Model architektúry nasadenia. [17]

7.1 Princíp detekcie

V prípade, že útočník napadne zraniteľný systém a spôsobí *buffer overflow*, jeho počínanie je detekované a zaznamenané v reálnom čase. Záznam komunikácie z TCPdumpu a paket, ktorý spôsobil útok sú poskytnuté extraktoru komunikácií, kde sú tieto údaje spracované. Z týchto dát, sú extrahované len relevantné údaje, ktoré sú následne poslané extraktoru metrík. Extraktor metrík vytvorí množinu metrík a uloží ju do databáze (Dataset). Množina metrík je potom zaslaná IDPS systému, ktorý si aktualizuje svoj model opakovaným učením [17].

Rovnako ako sa zaznamenávajú útoky, tak je potrebné robiť aj TCPdumpové záznamy legitímnych komunikácií. Model, ktorý bude uskutočňovať klasifikáciu komunikácií potrebuje mať dostatočné množstvo záznamov pre obe triedy. Preto by jeho trénovanie malo prebiehať vždy na všetkých záznamoch útokov a reprezentatívnej množine záznamov legitímnych komunikácií (ktorých je k dispozícii vždy viac ako záznamov o útokoch).

IDPS systém bude analyzovať všetku sieťovú prevádzku vo forme TCP komunikácií. Z každej TCP komunikácie budú extrahované metriky, ktoré sa následne klasifikujú natrénovaným modelom IDPS systému. Podľa kvality zaradenia do danej triedy dôjde v prípade útoku buď k odfiltrovaniu komunikácie jednoduchými firewallovými[1] pravidlami alebo k vygenerovaniu detekčnej správy upozorňujúcej na možný útok. Preto bude veľmi dôležité optimalizovať samotnú extrakciu metrík, tak aby bola uskutočňovaná čo najrýchlejšie, tzn. ideálne v reálnom čase.

7.2 Laboratórne podmienky a efektivita metrík

Navrhnuté metriky boli extrahované z dát zachytených v laboratórnych podmienkach, a preto nemohol byť využitý potenciál niektorých metrík. Ale na druhej strane mohlo dôjsť k využitiu takej vlastnosti metriky, ktorá by v reálnych podmienkach nebola pozitívne využitá na klasifikáciu útoku alebo legitímnej komunikácie. Tieto metriky alebo typy metrík s popisom odlišností od reálnych podmienok sú:

- CntOfNewFlows, cntOfOldFlows – počty novo vzniknutých spojení po začiatku analyzovaného spojenia alebo pred ním sú v laboratórnych podmienkach závislé len na komunikáciách emitovaných dvomi stanicami, ktoré sú navyše obmedzené počtom aj typom prevádzkovaných služieb,

- sumSessPerPort – počet TCP spojení v intervale +- 5 minút od aktuálneho spojenia, ktoré prebiehajú na rovnakom cieľovom porte. V simulovaných podmienkach opäť neexistoval kontext ostatných staníc, ktorý by bol analyzovateľný pre vyhodnotenie tejto metriky,

- niektoré dynamické, distribučné a behaviorálne metriky závislé na prenosových časoch jednotlivých paketov – reprezentácie rýchlostí prenosu paketov v oboch smeroch sú závislé počte prepínacích a smerovacích prvkoch medzi komunikujúcimi stranami a môžu byť tiež závislé na rýchlostiach sieťových kariet tak komunikujúcich strán ako aj prepínacích a smerovacích prvkov. Tiež sú závislé na použitej prípadnej dynamickej

[1] Analyzátor paketov a spojení filtrujúci sieťovú prevádzku na základe jednoduchých pravidiel. http://www.checkpoint.com/resources/firewall/.

ceste jednotlivých paketov, kde každý paket môže byť smerovaný inou cestou. V laboratórnych podmienkach sa s takýmito predpokladmi nepočítalo. Tento fakt bol uvažovaný pri návrhu metrík a je diskutovaný v sekcii č. 4.8,

- ratInoutB, ratInoutPkt, BytesTCPSess, BytesPerSessUpload, BytesPerSessDownload, Bytes3WH2FIN, CntAckIn, CntAckOut, všetky distribujúce a všetky behaviorálne aproximujúce – metriky, ktorých hodnoty sú priamoúmerne závislé na chybovosti prenosového kanálu a tým pádom aj na opätovnom prenášaní TCP paketov,

- všetky lokalizačné metriky – zdrojová a cieľová IP adresa a MAC adresa v laboratórnych podmienkach sú rovnaké a preto nemôžu byť uplatnené pri analýze aktuálneho spojenia. Taktiež nemôžu byť uplatnené ani metriky, ktoré udávajú či sa zdrojová alebo cieľová IP adresa nachádzajú v lokálnej sieti. Potenciál týchto metrík by bol využitý v prípade kedy by k aktuálnemu útoku došlo z prvotne napadnutej stanice nachádzajúcej sa v lokálnej sieti (napríklad v demilitarizovanej zóne[2]).

Preto sa v blízkej budúcnosti plánuje pre testovacie a ladiace účely nasadenie systému popísaného v kapitole 7 do reálnych podmienok verejnej siete. Ideálny kandidát na takýto typ siete je sieť obsahujúca stanice s množstvom rôznych operačných systémov a ich rozličných verzií, ktoré používajú rôzne úrovne zabezpečenia. V úvahu zatiaľ prichádza napríklad internátna sieť. Toto nasadenie prebehne pod záštitou projektu MPO AIPS bežiaceho na Fakulte Informačních Technologií v Brně.

[2]Fyzická alebo logická podsieť vystavujúca služby do nedôveryhodnej siete akou je Internet. http://www.techrepublic.com/article/solutionbase-strengthen-network-defenses-by-using-a-dmz/5756029.

Kapitola 8

Možnosti ďalšieho rozvoja

V prípade, že sa podarí úspešne nasadiť navrhnutý systém detekcie útokov vo verejnej sieti, sa môže pristúpiť k analýze nazbieraných dát. Na základe analýz sa následne vytvorí model, ktorý by bol schopný s požadovanými vlastnosťami klasifikovať jednotlivé komunikácie. Až po tomto kroku sa bude dať reálne vyhodnotiť skutočná detekcia *zero-day* útokov.

Výsledkom nasadenia môže byť aj zistenie nevhodnosti niektorých metrík. Prípadne sa budú musieť niektoré metriky prerobiť alebo normalizovať pre reálne podmienky.

8.1 Návrh ďalších metrík

Do úvahy pripadá aj návrh ďalších metrík. Zatiaľ sa zvažuje využitie operácií signálového spracovania na krivky, ktoré aproximujú dané spojenia alebo na samotné pakety spojenia, na ktoré sa dá pozerať ako na vzorky. Mohla by sa využiť konvolúcia signálov pre detekciu určitých rysov týchto priebehov. Mohlo by dôjsť aj k využitiu vektoru derivácí získaného z daných vzoriek. Tento vektor by sa ďalej mohol analyzovať a mohli by sa nad ním uskutočňovať podobné operácie extrakcie metrík ako nad samotnými vzorkami – aproximácie polynómami, Fourierové transformácie, produkty s funkciami, získanie distribučných vektorov a podobne.

8.2 Skvalitnenie detekčného modelu

Ďalším experimentálnym rozšírením súvisiacim so samotnou real time klasifikáciou komunikácií by mohla byť diskretizácia každej ordinárnej metriky na manuálne špecifikované intervaly podľa analýzy rozložení hustoty týchto atribútov, ktorá bola uskutočnená v sekcii 6.4. Použitý nástroj RapidMiner takúto formu diskretizácie nepodporuje a preto tento krok nebol uskutočnený. Problémom tohto prístupu by mohlo byť preučenie klasifikačného modelu na trénovaciu sadu dát a tento model by nebol schopný zovšeobecňovať.

Iným možným experimentálnym vylepšením je kombinácia výstupov jednotlivých modelov v snahe dosiahnuť presnejšiu klasifikáciu. Tieto výstupy by mohli byť váhované a v správnom pomere by mohli dosiahnuť lepšie výsledky ako v samostatných verziách.

8.3 Urýchlenie extrakcie metrík

Vzhľadom na potrebu uskutočňovania detekcie útokov v reálnom čase, bude nutné urýchliť proces samotnej extrakcie metrík všetkými dostupnými prostriedkami. Je možné zaviesť

paralelizáciu na úrovni niektorých výpočtov extrakcie jednotlivých metrík alebo tiež paralelizáciu, ktorá by využila možnosť extrakcie metrík po skupinách. Využitie hardwarovej implementácie metrík tiež pripadá v úvahu.

Ďalším potenciálnym vylepšením zameraným na urýchlenie, by mohlo byť zavedenie extrakcie len relevantných metrík, ktorých ohodnotenia sa najväčšou mierou podieľajú na celkovom výstupe výsledného modelu. Tento prístup by si vyžadoval aj priebežné vyhodnocovanie kvality ostatných metrík, ktoré by s novými typmi útokov mohli nadobudnúť podstatnejší význam z pohľadu celkovej detekcie útokov.

Kapitola 9

Záver

Podstatou tejto práce bolo navrhnutie takých metrík, ktorými by bolo možné detekovať čo najviac sieťových útokov vrátane *zero-day* útokov. Preto bol navrhnutý nástroj na extrakciu navrhnutých metrík z databáze naplnenej údajmi z TCPdump. Ďalej boli extrahované metriky analyzované a bol hľadaný vhodný klasifikačný model, ktorý sa zameriaval na správnu klasifikáciu útokov aj legitímnych komunikácií.

Úvodný krok tejto práce spočíval v analýze existujúcich princípov detekcie sieťových útokov, kde bol dôraz kladený najmä na princípy funkčnosti honeypotov a IDPS systémov. Bol tiež uvedený a popísaný reprezentatívny zástupca vysokointeraktívnych honeypotov Argos. Spôsob využitia tohto honeypotu bol uvedený v sieťovej architektúre nasadenia a tiež aj pri získavaní informácií o útokoch v implementačnej časti tejto práce. Na základe analýz existujúcich princípov detekcie sieťových útokov a výstupov, ktoré sú Argosom generované pri vzniku útoku, boli urobené ďalšie analýzy využiteľnosti týchto dát na rôznych úrovniach abstrakcie. Bola tiež analyzovaná dátová kolekcia KDD Cup 99 a boli diskutované jej nedostatky.

Hlavný dôraz tejto práce bol kladený na návrh nových metrík a ich kategorizáciu podľa charakteristických rysov a vlastností. Pri návrhu bolo stanovené, že metriky sa budú vzťahovať k TCP spojeniam s možnosťou uvažovania ich kontextu. Celkovo bolo navrhnutých 169 metrík, z ktorých niektoré sú výsledkom parametrizácie základných typov metrík. Na základe dodaných dát získaných pri simuláciách útokov a legitímnych komunikácií bola naimplementovaná aplikácia, ktorá extrahuje najprv spojenia a potom získa hodnoty všetkých navrhnutých metrík pre tieto spojenia – extraktor metrík.

Ako ďalší krok bola učinená analýza dátovej kolekcie KDD Cup 1999 v nástroji RapidMiner. V tomto kroku sa hľadal najlepší klasifikačný model schopný túto dátovú kolekciu správne klasifikovať. Ako najlepší model bol stanovený naivný Bayessovský klasifikátor, ktorý dosiahol úspešnosť správnej klasifikácie útokov v 95.77% prípadoch a v 0.77% prípadoch klasifikoval legitímne komunikácie ako útoky.

Po tejto analýze sa prešlo k experimentom zameraným na analýzu extrahovaných metrík z dát získaných z projektu MPO AIPS. Analyzovala sa najskôr hustota rozloženia navrhnutých metrík vzhľadom na príslušnosť do danej triedy. Bolo tu identifikovaných 45 metrík (z celkového počtu 881), ktoré vykazovali významné rozloženie hustoty prvkov vstupnej kolekcie. Tieto významné metriky sa následne použili pre experimenty zamerané na nájdenie vhodného klasifikačného modelu. Bolo to hlavne z dôvodu rýchleho generovania výpočtovo náročnejších konfigurácií modelov a tiež kvôli zameraniu klasifikačných metód na najvýznamnejšie metriky. Ako mierne významnejšie metriky z hľadiska smeru komunikácie boli identifikované tie, ktoré využívali odchádzajúce pakety. Ale zanedbateľný vplyv nemali ani

metriky využívajúce opačný smer.

Najprv sa experimentovalo s naivnou Bayessovou metódou, kde bola dosiahnutá úspešnosť správnej klasifikácie útokov v 91.76% prípadoch a v 12.72% prípadoch došlo ku nesprávnej klasifikácii legitímnych komunikácií ako útoky. Na základe analýzy korelačnej matice bolo ako ďalšie potenciálne vylepšenie navrhnuté použitie metódy PCA. Táto metóda však zlepšenie detekcie útokov nepriniesla. Úspešnosť správnej klasifikácie útokov bola dosiahnutá v 83.33% prípadoch a v 11.56% prípadoch došlo ku nesprávnej klasifikácii legitímnych komunikácií ako útoky. Neskôr sa pristúpilo k experimentom s diskretizáciou ordinárnych atribútov, kde boli najlepšie výsledky dosiahnuté pri binningu s použitím 5 intervalov. Bola tu dosiahnutá úspešnosť správnej klasifikácie útokov v 91.76% prípadoch, rovnako ako vo variante bez tejto metódy. Prínos použitia tejto metódy spočíva v redukcii nesprávne klasifikovaných legitímnych komunikácií. V tomto prípade je to len 5.78% čo je o vyše polovicu lepšie ako v predchádzajúcom experimente.

Jediný model, ktorý dokázal klasifikovať všetky útoky správne bol SVM. V SVM sa experimentovalo s typom kernelu a cenovým parametrom. Ako najvhodnejší kernel sa ukázal neurónový. V 100% prípadoch boli všetky útoky klasifikované správne lenže v 19.08% prípadoch boli legitímne komunikácie klasifikované ako útoky.

Pri použití modelu rozhodovacieho stromu sa podarilo nájsť konfiguráciu, kde dochádzalo k najlepšej klasifikácii legitímnych komunikácií v 97.69% prípadoch. Útoky tu boli správne klasifikované v 58.33% prípadoch. Tento model je vhodný do prostredí kde môže byť výpadok komunikácie siete alebo systému drahší ako samotné dôsledky útoku.

V nástroji RapidMiner bolo tiež experimentované aj s inými klasifikačnými modelmi, ale žiadny z nich nepriniesol dostatočne dobré výsledky. Zhrnutie dosiahnutých výsledkov jednotlivých modelov je uvedené v tabuľke č. 6.8. Vzhľadom na značný nepomer zástupcov jednotlivých tried klasifikácie, nie je údaj celkovej klasifikácie relevantný pre jednoznačné určenie kvality daného klasifikačného modelu.

Pre analýzu navrhnutých metrík bola použitá množina len 12 útokov a 173 legitímnych komunikácií a preto nemôžu byť výsledky rovnocenne porovnávané s výsledkami získanými pri dolovaní z dátovej kolekcii KDD Cup 1999, ktorá obsahuje rádovo desaťtisíce záznamov komunikácií. Na druhej strane nedostatkom metrík kolekcie KDD Cup 1999 je, že využívajú také znalosti, ktoré nie sú priamo získateľné z promiskuitného sledovania sieťovej prevádzky a takto získaných metainformácií o prebiehajúcich spojeniach. Tieto metriky využívajú analýzu obsahovej časti paketov (uvažujú dešifrovanie) a skúmajú dôsledok, ktorý na kompromitovanom systéme analyzované spojenie spôsobilo (počet neúspešných prihlásení, získanie prístupu k root shellu, počet vytvorených a pristupovaných súborov atď.).

Ďalej v rámci tejto práce bola diskutovaná architektúra nasadenia systému, ktorý by mal umožniť detekovať a zároveň predchádzať ako známym typom útokov, tak aj *zero-day* útokom. Bol tiež popísaný priebeh detekcie nového útoku a následná aktualizácia modelu IDPS systému. Tiež bol popísaný aj proces samotnej klasifikácie spojení sieťovej prevádzky.

Ako jeden z posledných krokov uskutočnených v tejto práci bolo diskutovanie vplyvu laboratórnych podmienok na potenciál jednotlivých skupín metrík. Sú to najmä metriky závislé na prenosových časoch jednotlivých paketov, na chybovosti prenosového kanálu a na absencii reálneho kontextu ostatných staníc. Zároveň je to väčšina lokalizačných metrík, ktoré využívajú IP adresy a MAC adresy komunikujúcich strán a taktiež príznaky výskytu komunikujúcich strán v lokálnej sieti, keďže môže ísť o útoky iniciované inou napadnutou stanicou umiestnenou v lokálnej sieti (napríklad v demilitarizovanej zóne).

Na záver boli rozobraté možnosti ďalšieho rozvoja navrhnutého systému, kde boli vyjadrené myšlienky súvisiace s návrhom ďalších metrík s využitím operácií spracovania signálov.

Boli tu tiež uvedené návrhy ako by sa dal skvalitniť klasifikačný model detekčného systému prostredníctvom špeciálnej varianty diskretizácie jednotlivých metrík alebo kombináciou viacerých klasifikačných modelov a následným váhovaním ich výstupov. Ďalšie návrhy súviseli s urýchlením samotného procesu extrakcie metrík a to formou paralelizácie výpočtov niektorých metrík alebo formou paralelnej extrakcie skupín metrík.

Keďže sa pracovalo v laboratórnych podmienkach, detekcia zero-day útokov nemohla byť posúdená. Preto ako nasledujúci krok je dôležité uskutočniť nasadenie honeypotov v reálnej sieťovej prevádzke a potom uskutočniť analýzu extrahovaných metrík.

Literatúra

[1] Confusion Matrix. [online], [cit. 12.5.2012].
URL http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html

[2] Naive Bayes Classifier. [online], [cit. 12.5.2012].
URL http://www.codeproject.com/Articles/318126/Naive-Bayes-Classifier

[3] RapidMiner wiki, Nominal to Numerical. [online], [cit. 12.5.2012].
URL http://rapid-i.com/wiki/index.php?title=Nominal_to_Numerical

[4] RapidMiner wiki, X-Validation. [online], [cit. 12.5.2012].
URL http://rapid-i.com/wiki/index.php?title=X-Validation

[5] Discretize by Binning. [online], [cit. 13.5.2012].
URL http://rapid-i.com/wiki/index.php?title=Discretize_by_Binning

[6] Stratified Sampling. [online], [cit. 13.5.2012].
URL http://www.coventry.ac.uk/ec/~nhunt/meths/strati.html

[7] RapidMiner wiki, Nominal to Numerical. [online], [cit. 20.5.2012].
URL http://rapid-i.com/content/view/181/190

[8] RFC 793 - Transmission Control Protocol. [online], [cit. 20.5.2012].
URL http://www.faqs.org/rfcs/rfc793.html

[9] Technical terms, malware. [online], [cit. 20.5.2012].
URL http://www.techterms.com/definition/malware

[10] Zero day attacks. [online], [cit. 24.12.2011].
URL http://www.bullguard.com/bullguard-security-center/security-articles/what-are-zero-day-attacks.aspx

[11] Metasploit framework. [online], [cit. 25.12.2011].
URL http://metasploit.com/about/what-is-it

[12] The NSL-KDD Data Set. [online], [cit. 25.12.2011].
URL http://www.iscx.ca/NSL-KDD

[13] Honeypot. [online], [cit. 26.12.2011].
URL http://www.securityfocus.com/archive/119

[14] Anagnostakis, K.; Sidiroglou, S.; Akritidis, P.; aj.: Shadow Honeypots. In *Proc. of the 14th USENIX Security Symposium*, 2005.

[15] Axelsson, S.: Intrusion detection systems: A survey and taxonomy. Technická zpráva, Technical report, 2000.

[16] Barabas, M.: Security Protocols for Wireless Networks - Malware detection . Technická zpráva.

[17] Barabas, M.; Drozd, M.; Hanacek, P.: Behavioral signature generation using shadow honeypot. *ICCNSS*, 2012.

[18] Fajmon, B.; Růžičková, I.: Matematika 3. [online], [cit. 12.1.2012].
URL http://www.umat.feec.vutbr.cz/~novakm/matematika3.pdf

[19] Galetka, J.: *Network attack analysis using honeypots*. Diplomová práce.

[20] Habra, N.; Charlier, B. L.; Mounji, A.; aj.: ASAX: Software Architecture and Rule-Based Language for Universal Audit Trail Analysis. In *ESORICS*, *Lecture Notes in Computer Science*, ročník 648, editace Y. Deswarte; G. Eizenberg; J.-J. Quisquater, Springer, 1992, ISBN 3-540-56246-X, s. 435–450.
URL http://dx.doi.org/10.1007/BFb0013912

[21] Jolliffe, I. T.: *Principal Component Analysis*. New York, NY, USA: Springer, 2002.

[22] Mathworld: Markov process. [online], [cit. 24.12.2011].
URL http://mathworld.wolfram.com/MarkovProcess.html

[23] Murata, T.: Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, ročník 77, č. 4, 1989: s. 541–580.

[24] Newsome, J.; Song, D.: Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software. In *NDSS05*, 2005.

[25] Provos, N.; Holz, T.: *Virtual Honeypots - From Botnet Tracking to Intrusion Detection*. Addison-Wesley, 2008, ISBN 978-0-321-33632-3, I–XXIII, 1–440 s.

[26] R. P. Lippmann, D.: Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. *discex*, ročník 02, 2000: str. 1012.

[27] S. J. Stolfo, W.: Cost- based modeling for fraud and intrusion detection: Results from the jam project. Technická zpráva, 2000.

[28] Stergiou, C.; Siganos, D.: Neural networks. [online], [cit. 1.1.2012].
URL http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

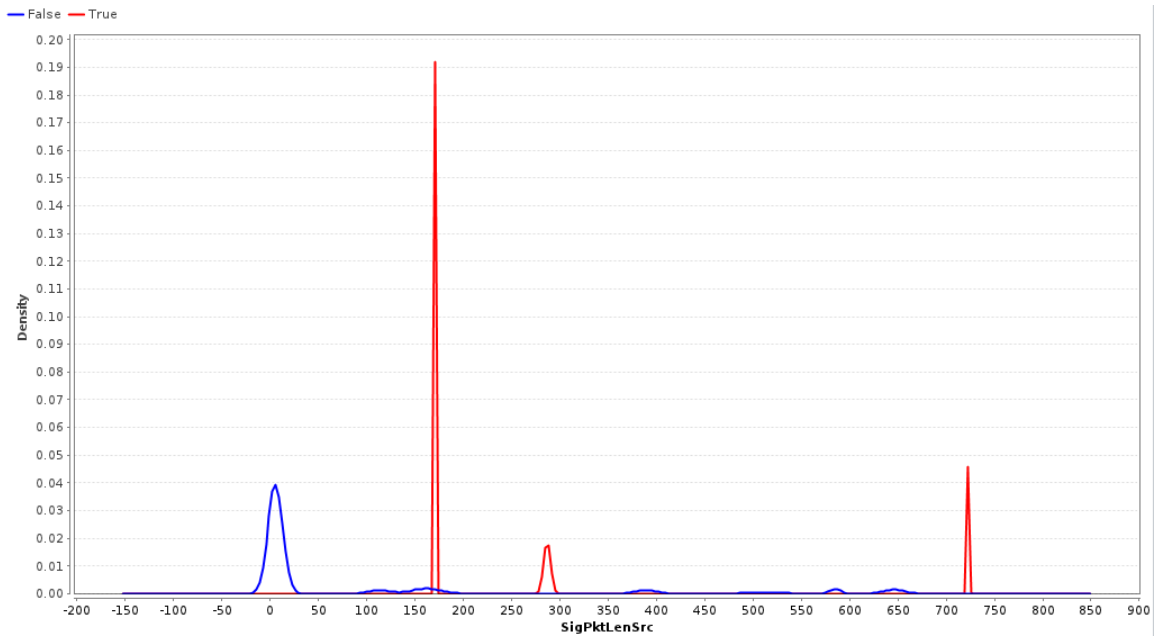
[29] Tavallaee, M.; Bagheri, E.; Lu, W.; aj.: A detailed analysis of the kdd cup 99 data set. In *Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*, 2009.

[30] University of California, I. U.: KDD Cup 99. [online], [cit. 1.3.2012].
URL http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

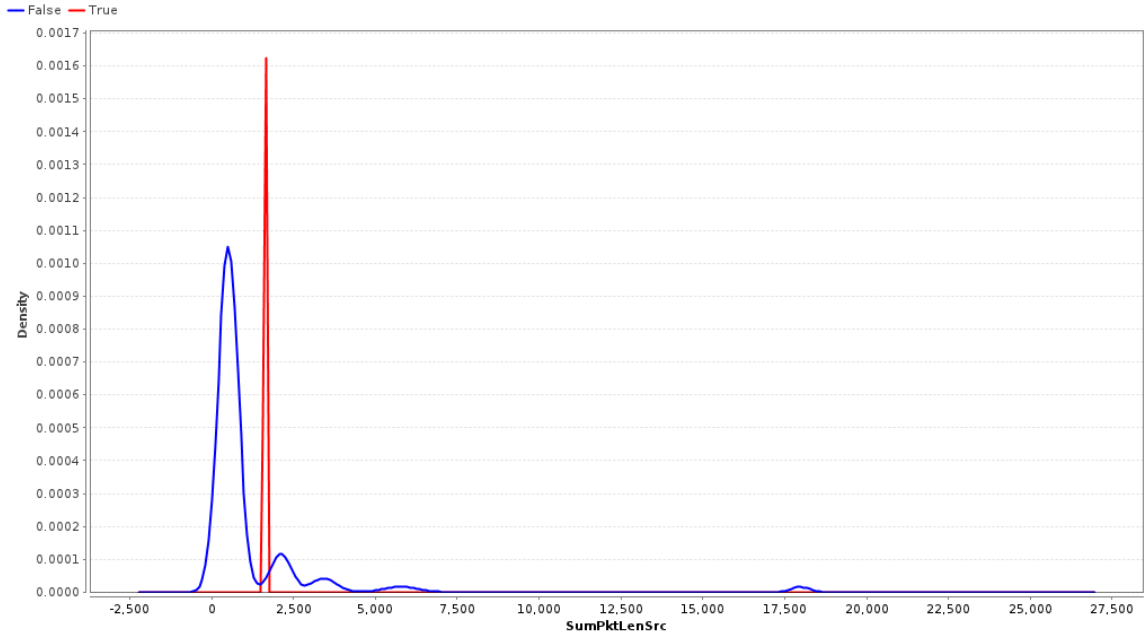
Dodatok A

Rozloženia hustoty metrík

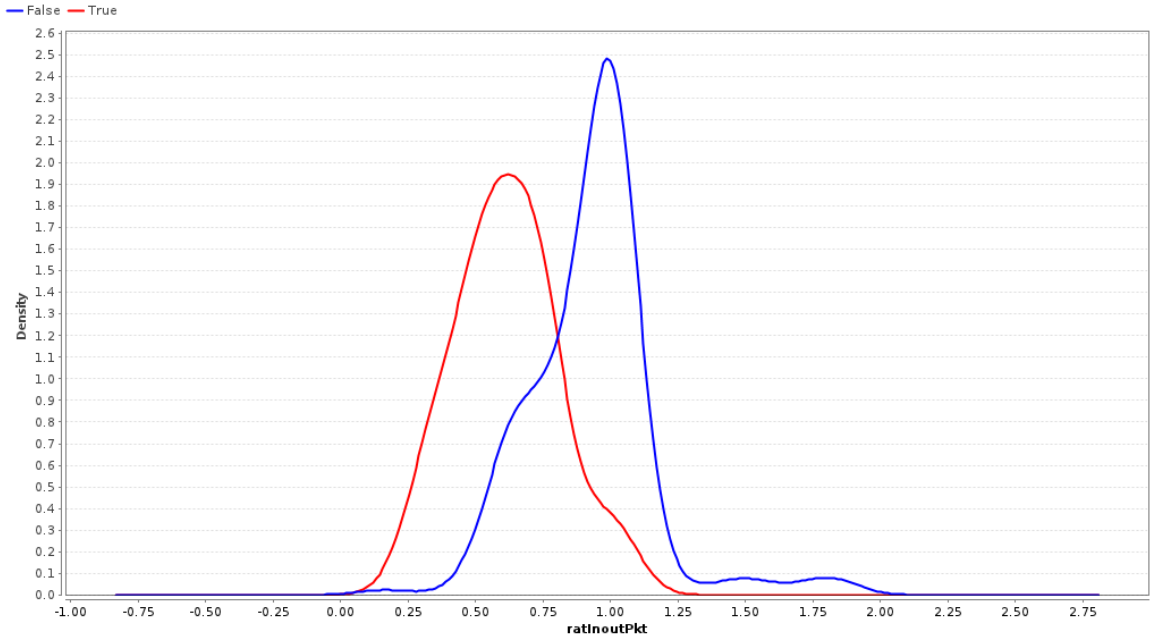
A.1 Štatistické metriky



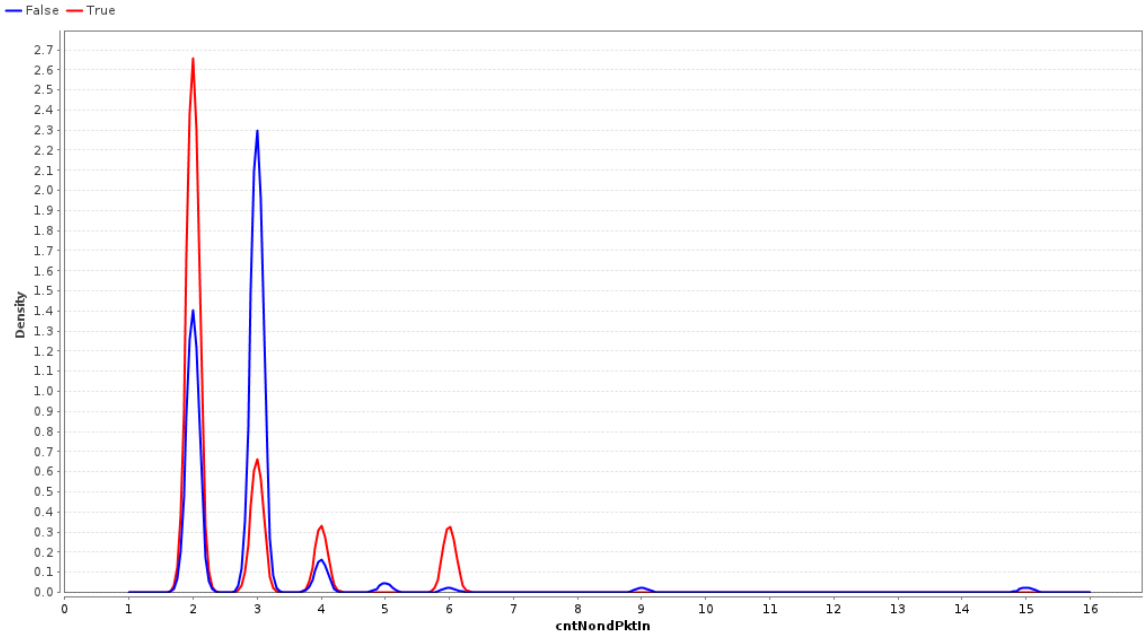
Obrázok A.1: Rozloženie hustoty metriky sigPktLenSrc.



Obrázok A.2: Rozloženie hustoty metriky sumPktLenSrc.

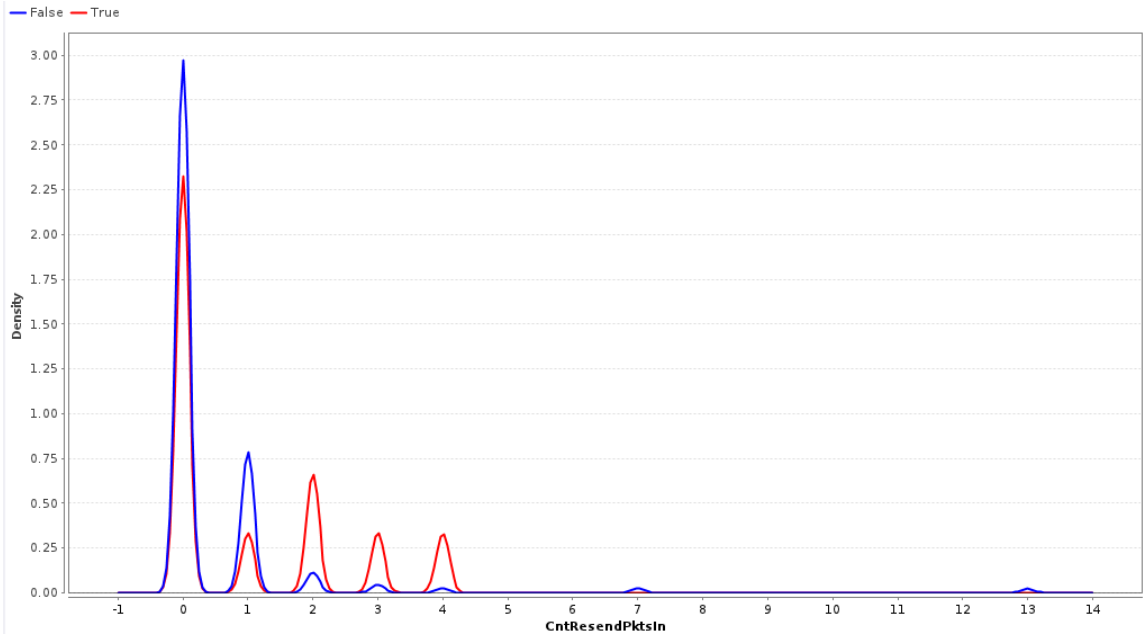


Obrázok A.3: Rozloženie hustoty metriky ratInOutPkt.



Obrázok A.4: Rozloženie hustoty metriky cntNondPktIn.

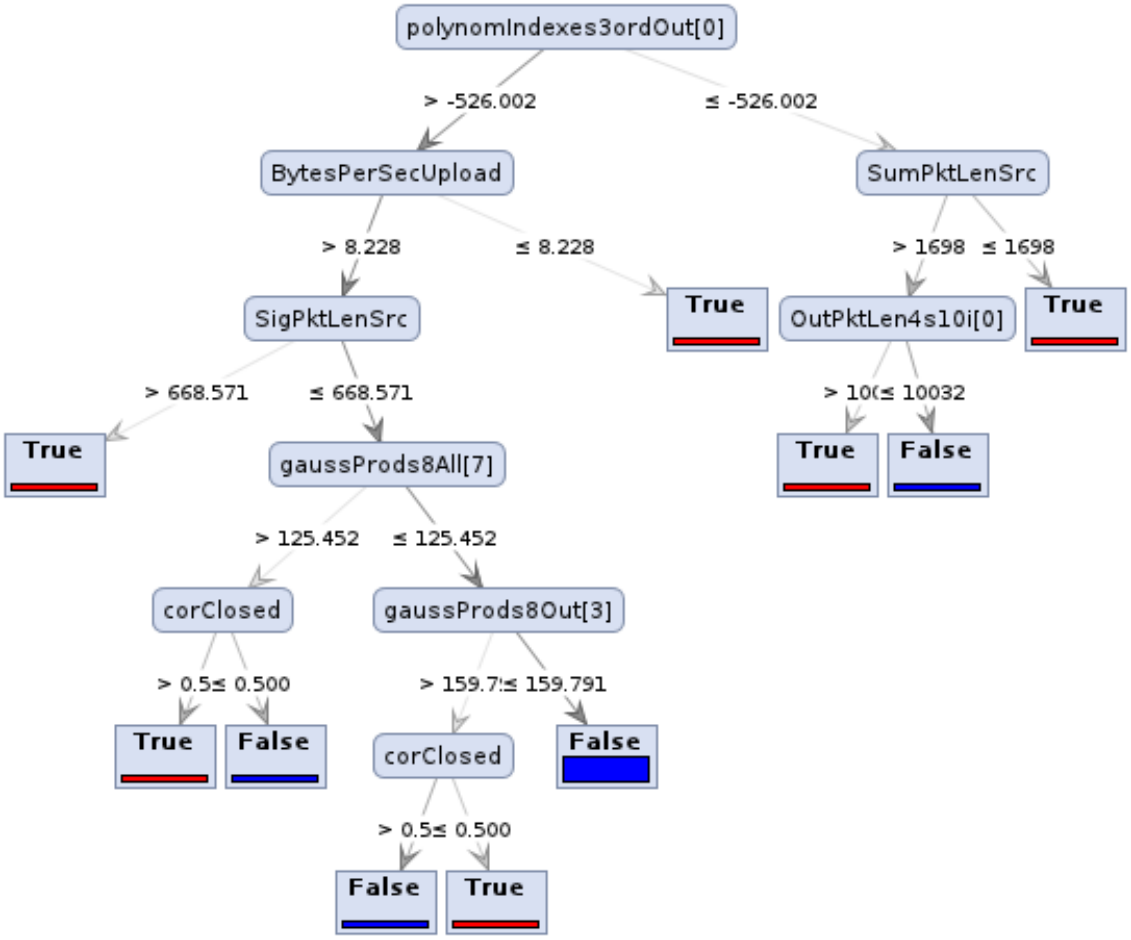
A.2 Dynamické metriky



Obrázok A.5: Rozloženie hustoty metriky cntResendPktsIn.

Dodatok B

Rozhodovací strom



Obrázok B.1: Grafická reprezentácia nájdeného rozhodovacieho stromu.

Dodatok C

Obsah DVD

- **doc** - zdrojové kódy LaTeX-u a lyx-u textovej časti diplomovej práce,
- **outputs** - výstupy generované nástrojom na extrakciu metrík,
- **rapidMiner** - zdrojové kódy použitej verzie nástroja rapidMiner a tiež uložené konfigurácie procesov, pri ktorých boli dosiahnuté zaujímavé výsledky,
- **src** - zdrojové kódy nástroja na extrakciu metrík,
- **README[ENG]** - popis obsahu priloženého nosiča v anglickom jazyku,
- **README[SVK]** - popis obsahu priloženého nosiča v slovenskom jazyku.