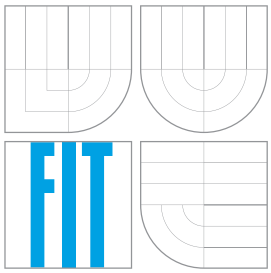
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

BEZPEČNOSTNÍ ANALÝZA SÍŤOVÉHO PROVOZU POMOCÍ BEHAVIORÁLNÍCH SIGNATUR

SECURITY ANALYSIS OF NETWORK TRAFFIC USING BEHAVIORAL SIGNATURES

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE
AUTHOR

Ing. MAROŠ BARABAS

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Dr. Ing. PETR HANÁČEK

BRNO 2016

Abstrakt

Tato práce se zaměřuje na popis aktuálního stavu bádání v detekčních metodách síťových útoků a následně na vylepšení schopnosti detekce specifických útoků vytvořením formálního popisu síťových metrik, které aproximují průběh síťového spojení a vytváří signaturu založenou na behaviorální charakteristice analyzovaného spojení. Cílem práce není prevence vůči aktuálně probíhajícím útokům ani reakce na tyto útoky, důraz se klade na analýzu spojení, získání co největšího množství informací a vytvoření základu detekčního systému, který dokáže minimalizovat velikost dat sbíraných ze sítě při ponechání nejdůležitějších informací pro následující analýzu. Hlavním cílem práce je vytvoření konceptu detekčního systému, který pomocí definovaných metrik redukuje síťový tok na signatury spojení s důrazem na behaviorální aspekty komunikace. Koncept zvyšuje autonomnost detekčního systému pomocí vytvoření expertní znalosti z honeypot systému s podmínkou nezávislosti na technologických aspektech analyzovaných dat (např. šifrování, použité protokoly, technologie nebo prostředí). Použití konceptu expertní znalosti honeypot systému v roli učitele klasifikačních algoritmů vytváří autonomnost systému při detekci neznámých útoků. Dále nabízí možnost samostatného učení (bez zásahu člověka) na základě poznatků získaných z útoků na tyto systémy. V práci je představen postup vytvoření laboratorního prostředí a experimenty s definovanou signaturou spojení nad získanými daty i nad převzatou testovací databází. V závěru jsou porovnány dosažené výsledky s aktuálním přehledem síťových detekčních systémů a je vyzdvižen přínos navržených metod aproximujících průběh analyzovaného spojení.

Abstract

This thesis focuses on description of the current state of research in the detection of network attacks and subsequently on the improvement of detection capabilities of specific attacks by establishing a formal definition of network metrics. These metrics approximate the progress of network connection and create a signature, based on behavioral characteristics of the analyzed connection. The aim of this work is not the prevention of ongoing attacks, or the response to these attacks. The emphasis is on the analysis of connections to maximize information obtained and definition of the basis of detection system that can minimize the size of data collected from the network, leaving the most important information for subsequent analysis. The main goal of this work is to create the concept of the detection system by using defined metrics for reduction of the network traffic to signatures with an emphasis on the behavioral aspects of the communication. Another goal is to increase the autonomy of the detection system by developing an expert knowledge of honeypot system, with the condition of independence to the technological aspects of analyzed data (e.g. encryption, protocols used, technology and environment). Defining the concept of honeypot system's expert knowledge in the role of the teacher of classification algorithms creates autonomy of the system for the detection of unknown attacks. This concept also provides the possibility of independent learning (with no human intervention) based on the knowledge collected from attacks on these systems. The thesis describes the process of creating laboratory environment and experiments with the defined network connection signature using collected data and downloaded test database. The results are compared with the state of the art of the network detection systems and the benefits of the proposed approximation methods are highlighted.

Klíčová slova

aproximace, behaviorální signatura, bezpečnost, detekce, honeypot, IDS, klasifikace síťového provozu, metriky, přetečení zásobníku, síťová analýza, síťové útoky

Keywords

aproximation, behavioral signature, buffer overflow, detection, honeypot, IDS, metrics, network analysis, network attacks, network traffic classification, security

Citace

Maroš Barabas: Bezpečnostní analýza síťového provozu pomocí behaviorálních signatur, disertační práce, Brno, FIT VUT v Brně, 2016

Bezpečnostní analýza síťového provozu pomocí behaviorálních signatur

Prohlášení

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně pod vedením pana doc. Dr. Ing. Petra Hanáčka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Maroš Barabas
12. apríla 2016

Rád by som túto prácu venoval všetkým, ktorí mali vieru a trpezlivosť, tým, ktorí prispeli radami a snahou a hlavne tým, ktorí boli mojou motiváciou a oporou, ktoré ma viedli k cieľu.

Obsah

Zoznam obrázkov

Zoznam tabuliek

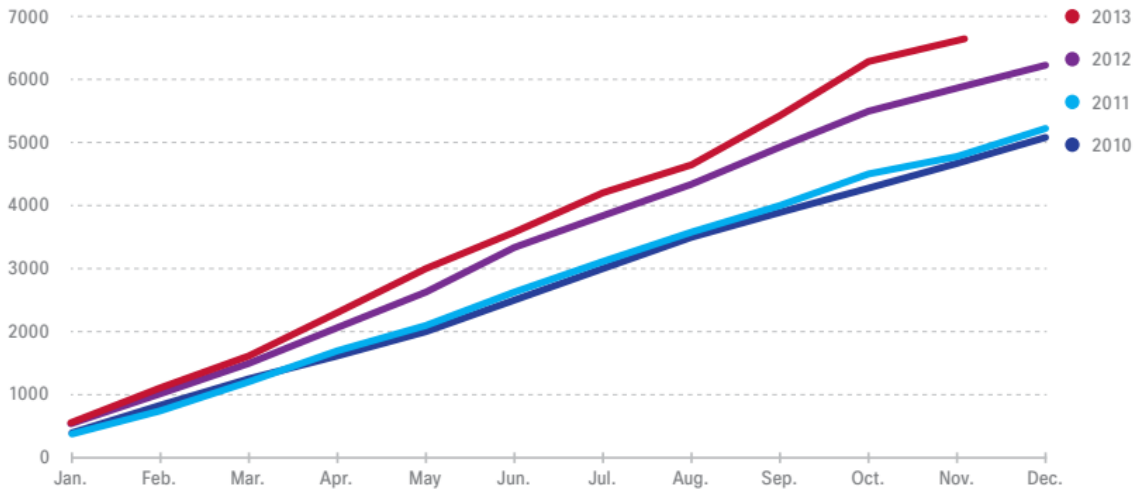
Kapitola 1

Úvod

Internet v dnešnej dobe zasahuje do každodenného života viac ako 40 % svetovej populácie [87, 31]. Aktuálne smery vývoja informačných technológií, vznikajúca globálna informačná architektúra založená na Internete uľahčujúca výmenu služieb a tovaru (známa i ako *Internet of Things* alebo *Internet of Everything*, skrátene *IoT*) [139] spôsobujú, že bežné zariadenia v našom prostredí sú pripájané do Internetu. Podľa predpovedí spoločnosti Gartner z roku 2014 je technológia *IoT* na vzostupe a očakáva sa do 5 až 10 rokov jej komerčný rozmach [52]. Tento trend sa postupne dostáva i do rôznych odvetví ako automobilový priemysel alebo bežné doplnky – chytré telefóny, hodinky, šperky, či dokonca i chytré oblečenie[1], ale napríklad i letecký priemysel pre ktorý je dôležitým prvkom pripojenie pasažierov k zábavnému systému lietadla a k Internetu. Ďalšou veľkou oblasťou je priemyselná automatizácia, tzv. *ICS* systémy (*Industrial Control Systems*), ktoré prenikajú do oblasti pripájania riadiacich súčastí priemyselných systémov do siete a ich globálne riadenie, napr. *SCADA* (*Supervisory Control and Data Acquisition*) alebo *Smart Grid* systémy [42, 122]. Trend automatizácie systémov ovplyvňujúcich fyzický svet a ľudské prostredie sa postupne prenáša do miest (tzv. *Smart City* – smer využitia ICT technológií na skvalitnenie života v mestách), i do ľudských príbytkov (tzv. *Home Automation* – automatizácia systémov riadiacich prostredie domov alebo bytov, ako napr. vykurovanie, vzduchotechnika, osvetlenie, apod.). Trend pripájania vecí, ktoré majú vplyv na fyzické prostredie do Internetu so sebou prináša určité hrozby, napr. v podobe získavania informácií z týchto zariadení alebo ich neoprávnené ovládanie. Vzhľadom na povahu takýchto systémov môže mať ich prípadné zneužitie dopad i na bezpečnosť človeka a prostredia, napríklad pri prípadnom útoku na systém automobilu a ovplyvnenie jeho jazdných vlastností [88]. Ako je vidieť na obrázku 1.1 (os y: počet incidentov v danom mesiaci; zdroj: CISCO [33]), medzi rokmi 2010 a 2013 zaznamenala spoločnosť CISCO nárast zaznamenaných bezpečnostných hrozieb a zraniteľností o približne 12 % [33]. V bezpečnostnej správe od spoločnosti Secunia [117] sa uvádza nárast zaznamenaných zraniteľností v roku 2014 o 18 % oproti roku 2013 (celkový počet zraniteľností 15 435) a 55 % nárast v rámci rokov 2009–2014.

S postupujúcimi technológiami sa mení i charakter hrozieb a odpoveďou na tieto hrozby je zvýšený záujem o bezpečnosť. Na základe výskumu spoločnosti Gartner, sa očakáva do roku 2020 zvýšenie financií alokovaných na okamžitú detekciu a reakciu na bezpečnostné hrozby na 75 % (z 10 % v roku 2012). Ako sa mení charakter informácií a zdrojov dostupných prostredníctvom Internetu, mení sa i povaha útočníkov. Obecný pohľad na útočníkov

[1] Pre viac informácií pozri URL: https://www.wearable-technologies.com/2015/03/a-look-at-smartclothing-for-2015/



Obr. 1.1: Kumulatívna štatistika zobrazujúca nárast incidentov.

ako na osamelých mužov žijúcich v podzemí, ktorí pomocou svojho počítača žijú vo virtuálnom prostredí a ovládajú sieť ovládnutých počítačov [138], dnes vznikajú nové formy útočníkov, ako sú tzv. hacktivisti [71] – politicky motivovaní útočníci, ktorí sa na základe svojho politického presvedčenia snažia diskreditovať alebo inak ublížiť proti-strane alebo jednotlivcovi. Motivácia útočníkov môže byť ale vyvolaná i vidinou zisku alebo osobnej slávy v rámci hackerskej komunity. Vidina zisku tak otvára možnosti čierneho trhu, ktoré využívajú skupiny hackerov k predaju informácií alebo vlastných služieb za peniaze. Jedným z často sa opakujúcich pojmov, poslednú dekádu, sú pokročilé perzistentné hrozby (z angl. *Advanced Persistent Threats*), alebo *APT* [123], ktoré označujú sofistikované techniky a vektory útokov, ktoré slúžia na obchádzanie bezpečnostných mechanizmov, skrytie činností útočníka a jeho nedetegované zotrvanie na napadnutých systémoch v čo najdlhšom čase, zväčša za účelom získania informačných aktív obete. Ako *APT* bývajú často označované útoky, ktoré sú sponzorované štátnymi zložkami, ale toto tvrdenie je ťažké dokázať [68]. Vytvorenie týchto techník je zložité a útočníci sa tak presúvajú z pozície jednotlivca do organizovaných skupín, ktoré môžu byť financované firmami, ale napríklad i vládami za účelom špionáže (často označovanej i ako kyber-šiponáž, z angl. *Cyber-espionage*) alebo za účelom útoku na iný štát s deštruktívnym úmyslom narušenia, prípadne zničenia (tiež známe pod pojmom *Cyberwar* alebo *Cyberwarfare*) [41].

V posledných rokoch sa objavujú stále častejšie zverejňované rozsiahle útoky, ktoré je možné označiť ako *APT*. Nižšie uvedené príklady útokov majú spoločné charakteristiky – sú cielené, sú sofistikované a ako možní útočníci sú väčšinou označované vlády, prípadne skupiny pracujúce pre vládne organizácie rôznych štátov.

Stuxnet. Jedným z prvých útokov v histórii *APT* bol malware *Stuxnet*, ktorý bol zameraný na *ICS*, konkrétne *SCADA* systémy [76], útočil na riadiace systémy od spoločnosti Siemens a pri napadnutí zariadenia menil frekvenciu jeho otáčania. Špekulácie uvádzajú, že týmto správaním bol pravdepodobne určený na zničenie centrifúg Iránskych nukleárnych zariadení [128]. Vzhľadom na komplexnosť a sofistikovanosť tohoto malware sa predpokladá, že za prípravu a financovanie je zodpovedná vláda USA (nikdy to ale nebolo dokázané) [77]. Ďalšie útoky, ktoré boli odhalené ukazujú na mieru veľkosti skupín, ktoré za nimi stoja.

Red October. V októbri roku 2012 [113], po sérii útokov na počítačové siete rôznych medzinárodných diplomatických agentúr, spoločnosť Kaspersky v priebehu vyšetrovania objavila veľkú kyber-špionážnu sieť pozostávajúcu z viac ako 60 C&C serverov[2] a domén hlavne v Nemecku a Rusku. V rámci analýzy boli nájdené rôzne použité exploity[3] (na zraniteľnosti CVE-2009-3129, CVE-2010-3333, CVE-2012-0158 a iné), niektoré využívajúce útok pretečením zásobníka (z angl. *Buffer Overflow*), všetky mierili na zraniteľnosti v balíkoch Microsoft® Office. Cieľom útokov boli veľvyslanectvá vo viac ako 50 krajinách.

Titan Rain. Od roku 2007 boli napadnuté viaceré britské a americké vládne úrady[95]. Podľa vyjadrenia Ministerstva obrany Veľkej Británie (Angl.: *The Ministry of Defense, MoD*) stojí za útokmi Čínska hackerská skupina (viaceré zdroje uvádzajú, že skupina je vedená pod *ČĽA*[4]).

Ďalšie príklady kybernetických útokov. V apríli 2011 unikli z japonskej spoločnosti **Sony Corporation** (konkrétne išlo o službu *PlayStation Network*) osobné údaje približne 100 miliónov užívateľov [22]. V apríli 2011 unikli mená a emailové adresy z marketingovej spoločnosti **Epsilon**, ktorá poskytuje emailové kampane na viac ako 40 mld. emailov ročne pre viac ako 2 500 spoločností [59]. **Flame**[5] [56] je jeden z najsofistikovanejších malware, ktoré boli odhalené. Malware funguje ako špionážny nástroj, ktorý dokáže nahrávať audio (napr. hovory zo Skypu), zaznamenávať obrazovku, funguje i ako *keylogger*[6] a dokáže zaznamenávať sieťovú činnosť obete. Všetky nazbierané informácie sú následne odosielané vlastníkovi (zväčša útočníkovi). Jedným z najzávažnejších útokov posledných rokov bol útok na kontraktorov armády USA, medzi ktorými bola napadnutá i spoločnosť **Lockheed Martin**. V prvom kroku bola napadnutá spoločnosť *RSA*, ktorá vyrába bezpečnostnú technológiu *Secure ID* pre viac-faktorovú autentizáciu, informácie z tohoto útoku boli následne využité k prieniku do Lockheed Martin, Boeing a ďalších spoločností, ktoré obsahujú citlivé informácie o zbraňových systémoch a vojenských technológiách [50].

1.1 Motivácia a cieľ práce

V predchádzajúcej kapitole práce boli stručne uvedené aktuálne problémy v oblasti bezpečnosti informačných systémov (v kapitole 1.4 sú uvedené základné informácie o sieťových bezpečnostných technológiách, ktoré sa snažia adresovať tieto problémy analýzou sieťového toku, prípadne nalákaním útočníka na nezabezpečené sieťové služby a analýzou jeho správania). Myšlienka vylepšenia návrhu detekčných systémov, ktoré by dokázali detekovať uvedené útoky vznikla okolo roku 2009 pri spustení projektu Automatizované zpracování útoků (FR-TI1/037 financovaný MPO ČR). Cieľom tohto projektu bol výskum a vývoj nových metód pre automatickú detekciu útokov a malware. Výskum sa začal neskôr uberať

[2] C&C – z angl. *Command and Control*, je systém dostupný z Internetu pre komunikáciu so SW (napr. malware, prípadne *rootkity* apod.) útočníka. Infikovaný stroj nemusí byť dostupný z Internetu a preto je tento server v určitých intervaloch kontaktovaný na zabezpečenie prístupu útočníkovi a na vykonávanie príkazov.

[3] Exploit – software alebo nástroj, ktorý je navrhnutý na využitie slabiny alebo zraniteľnosti v cieľovom systéme (typicky pre napadnutie systému, inštaláciu malware apod.)

[4] Čínska ľudová oslobodzovacia armáda (Angl.: *People's Liberation Army, PLA*) je názov vojenských síl Čínskej ľudovej republiky.

[5] Flame malware je často označovaný ako *w32.Flame.skywiper*

[6] Keylogger je nástroj na zaznamenávanie aktivity klávesnice, býva často spojený s nelegálnou činnosťou (súčasť malware) a špionážou, používa sa na získanie citlivých údajov akými sú prístupové údaje, čísla kreditných kariet apod.

k detekcii útokov *buffer overflow* s využitím honeypot systémov na automatické generovanie signatúr pre sieťový detekčný systém[7].

1.1.1 Motivácia

Ako bolo poukázané v úvode kapitoly 1, aktuálny stav a trendy v bezpečnosti poukazujú na neschopnosť efektívne čeliť moderným kybernetickým útokom, odolnosť aplikácií, systémov, či infraštruktúr voči kybernetickým útokom je na stále nedostatočnej úrovni. Tento fakt vyplýva nielen z uvedených článkov popisujúcich bezpečnostné incidenty, ale i z rôznych renomovaných štúdií, ktoré sa zaoberajú vývojom kybernetickej bezpečnosti. Ako príklad je možné uviesť správy technologických spoločností zaoberajúcich sa počítačovou bezpečnosťou, ako McAfee, Symantec, Bit9, Cisco a ďalších [33], ktoré spoločne upozorňujú nielen na vysoký nárast počtu nových útokov, ale predovšetkým na ich sofistikovanosť a zameranie na neustále nové ciele a technológie. Poznatky z týchto správ je možné zhrnúť nasledovne:

- Informačné systémy sú architektonicky navrhnuté a postavené s cieľom plniť požadované funkcie bez zamerania na bezpečnosť.
- Rapídne pribúda počet sofistikovaných malware a vektorov útokov, čiastočne sa spomaľuje pribúdanie zraniteľností a útočníci sa viac zameriavajú na mobilné platformy.
- Pokročilé počítačové útoky využívajú neznáme zraniteľnosti a vlastnosti systémov k nelegálnej aktivite.
- Bezpečnostní špecialisti vedia o vysokej miere rizika, ale nie sú spravidla schopní ich riešiť s prideleným rozpočtom a dostupnými technológiami.
- Kybernetické útoky sú doménou organizovaných skupín s veľmi vysokým know-how a motiváciou.
- Objavujú sa nové typy útočníkov – napr. hacktivisti, ktorí neútočia pre zisk, ale pre politickú motiváciu.

Podstata kybernetických útokov sa prispôsobuje aktuálnym trendom, ktoré prinášajú viac nezabezpečených zariadení do Internetu (tendencia byť „online“, mať čo najviac zariadení pripojených do siete), zapojenie personálnych mobilných zariadení do podnikového prostredia, presun kritických služieb (napr. bankové a platobné služby) bližšie k zákazníkom – na Internet, príchod internetových obchodov, ponuka komodít a služieb cez Internet (hudba, filmy), príchod cloudových služieb atď. Ciele útokov sa menia z anonymných zraniteľných systémov spoločností na konkrétnych ľudí, do popredia vystupuje sociálne inžinierstvo, ako počiatočný vektor útoku, útočí sa na personálne počítače zamestnancov, odkiaľ je následne vedený ďalší útok.

U pokročilých foriem útokov boli spozorované [113, 95, 22, 59, 56, 50] veľmi sofistikované formy útokov (malware), ktoré boli prispôsobené na mieru obete/cieľa a zväčša využívali neznáme zraniteľnosti v systémoch (tzv. *zero-day*[8] zraniteľnosti). V prípade, že moderné

[7] Autor tejto práce pôsobil na projekte ako spoluriešiteľ so zameraním na vytvorenie nových metód automatickej detekcie, definície metrík sieťového toku a vytvorenie systému pre detekciu útokov na honeypot systém s možnosťou automatického generovania definovaných signatúr do detekčného systému.

[8] *Zero-day* alebo *0-day* zraniteľnosti je označenie zraniteľností, ktoré nie sú v danom momente známe a neexistuje pre nich záplata. *Zero-day exploit* je adekvátny pojem pre program zneužívajúci zero-day zraniteľnosť pre definovanú činnosť, zväčša prebratie kontroly nad činnosťou napadnutého programu.

sofistikované útoky, ktoré sú vedené štátmi organizovanými skupinami cielené zväčša na aktíva iných významných skupín (iné štáty, nadnárodné organizácie, prípadne organizácie, ktoré slúžia ako medzi krok pre získanie aktív, ktoré sú následne použité k napadnutiu vybraného cieľu), označujú sa ako *APT*.

Obrana voči takýmto útokom nie je jednoduchá, ale i napriek tomu existuje niekoľko komerčných nástrojov a technológií, ktoré sa snažia detegovať, prípadne zastaviť útoky na chránenú infraštruktúru a systémy. Dnešné bezpečnostné systémy sú založené prevažne na nasledujúcich princípoch:

- signatúry známych útokov vytvorené zo zachytenej komunikácie na zraniteľné systémy;
- manuálne definované vzory správania validnej komunikácie alebo útokov a ich vzájomné odlíšenie;
- štatistické vyhodnotenie analyzovaných sieťových parametrov s vyhľadávaním odchyliek;
- komunikácia na IP adresy, ktoré sa nachádzajú na zoznamoch (tzv. *black-listoch*) infikovaných adries;
- hybridné systémy využívajúce kombinácie uvedených princípov;

Nevýhoda týchto princípov spočíva v ich predikovateľnosti, kedy útočník so znalosťou nasadeného detekčného systému dokáže pripraviť útok na mieru a vyhnúť sa detekcii. Jedná sa napríklad o časové rozprestretie útoku (vkladané náhodne veľké oneskorenia odosielania paketov apod.), obfuskácia[9] binárnej reprezentácie útoku (zabalenie samotného *exploitu* a jeho rozbalenie na infikovanom systéme, náhodné a cielené zmeny v kóde, prehadzovanie častí kódov apod.), vkladanie šumu a náhodnosti do komunikácie, využívanie známych protokolov (napr. maskovanie za HTTP komunikáciu, tunelovanie protokolov, využívanie často povoľovaných portov apod.). Čoraz častejšie je v rámci komunikácie infikovaného počítača (malware, ktorý sa hlási na C&C server pre stiahnutie príkazov, modulov, aktualizácií apod.) preferovaný zabezpečený šifrovaný kanál (napr. HTTPS) a to z dôvodu prípadnej analýzy dátovej časti paketu a detekcie na základe odchytenia podozrivej komunikácie.

Z uvedených dôvodov je dôraz pri navrhovaní detekčného systému kladený na popis komunikácie na základe metrík, ktoré zo vstupných parametrov zachyteného toku dát vytvára profil komunikácie odzrkadľujúci určité správanie. Pri skúmaní správania človeka je možné pozorovať vytváranie určitých vzorov (rôzne aspekty vykonávania určitých aktivít, zvyky, nadväznosti udalostí a charakterové črty) a v prípade pozorovania odchyliek vo vytvorených vzoroch je možné určiť primeranú reakciu. Tak isto môžeme túto profiláciu preniesť do správania na Internete/v spoločnosti/na analyzovanom počítači, modelovať vzory správania a prípadné odchýlky prenášať do adekvátnych (i automatických) reakcií, napr. zamedzenie komunikácie pre podozrenie z infekcie. Tento princíp je možné použiť pri modelovaní komunikácie (napr. komunikácia pomocou nestavového protokolu HTTP bude vykazovať črty, ktoré pri komunikácii video-hovoru nebudeme pozorovať a naopak). Tým je možné naučiť systémy na rozpoznávanie určitých protokolov – vzorov správania a pri ich odchýlke (napr. tunelovanie komunikácie malware v rámci HTTP protokolu) môže byť podozrivá komunikácia označená pre ďalšiu analýzu. Tento princíp je možné otočiť pri učení

[9]Obfuskácia je zámerné zneprehľadnenie (napr. kódu, správy alebo informácie) za účelom znemožnenia porozumenia, či skrytia obsahu alebo účelu.

vzorov zo zachytených útokoch. V prípade, že bude tento princíp použitý na zariadenie alebo užívateľa, je možné hľadať vzory v správaní na vyššej (snáď abstraktnejšej) úrovni, napr. čas príchodu do práce, nadväznosť udalostí – zapnutie počítača, spustenie webového prehliadača, prihlásenie do sociálnych sietí, pauza (napr. na cigaretu) apod. Je zrejmé, že simulácia správania konkrétneho užívateľa je z pohľadu útočníka veľmi náročná až nemožná (pri neznalosti profilu užívateľa) na rozdiel od simulovania určitého protokolu alebo komunikácie.

V prípade, že bude detekcia obmedzená na modelovanie správania konkrétnych zariadení a komunikácií, je potrebné sa zamerať na správanie malware a útočníkov pri samotnej infekcii. V rámci práce [91], Moore ukázal, že malware sa šíri exponenciálnou radou až kým nedôjde k saturácii v danom segmente siete a filtrácia pomocou signatúr na perimetri siete dosahuje oveľa lepšie výsledky v prípade rýchlej aktualizácie signatúry útoku. Tento fakt priniesol myšlienku vytvárania signatúr pre detegované útoky v skoro reálnom čase tak, aby sa infekcia zastavila na prestupoch medzi segmentami siete a nedochádzalo k ďalšiemu rozšíreniu infekcie. Pre automatickú reakciu na útok v podobe vytvorenia signatúry je vhodné použitie honeypot systémov, ktoré dokážu analyzovať prichádzajúci útok zo siete a pri vytvorení signatúry je jej distribúcia na perimeter siete skoro okamžitá. Zároveň tieto systémy vytvárajú návnadu pre útočníka, čím zvyšujú pravdepodobnosť, že pri prvotnom napadnutí nedôjde k ohrození skutočných aktív spoločnosti. Cieľom je tak vytvorenie komplexnejšieho detekčného systému, ktorý je schopný detekcie na základe analýzy správania (na rôznych úrovniach), i okamžitej reakcie na nepoznaný útok v prípade napadnutia honeypot systému.

1.1.2 Ciele práce

Hlavným cieľom práce je vytvoriť sieťový detekčný systém pre paketové siete, ktorý pomocou definovaných metrík redukuje sieťový tok na signatúru spojenia s minimálnou stratou informácií. Cieľom práce nie je prevencia voči aktuálne prebiehajúcim útokom, ani reakcia na tieto útoky, dôraz sa kladie na analýzu spojenia, získania čo najviac informácií a vytvorenie základu detekčného systému, ktorý dokáže minimalizovať veľkosť dát zbieraných zo siete s ponechaním najdôležitejších informácií pre nasledujúcu analýzu.

Požiadavky na detekčný systém

Normálne správanie obecne poukazuje na množinu charakteristík, ktoré je možné pozorovať a extrahovať pri bežnom fungovaní pozorovaného zdroja. Nevalidné je potom také, ktoré vybočuje z bežného modelu fungovania tým, že vykazuje anomálie v pozorovaných charakteristikách. Počínajúc touto definíciou je možné nastaviť základné požiadavky na detekčný systém, ktorý bude rozlišovať normálne (validné) a nevalidné (útok) správanie v rámci analyzovanej (pozorovanej) siete. Pri zachovaní čo najväčšej všeobecnosti navrhnutého systému, budeme uvažovať, že zdroje charakteristík sú koncové body siete (niekedy označované ako uzly, stanice alebo aktéri analyzovanej komunikácie) i za predpokladu, že extrakcia charakteristík nebude prebiehať na týchto uzloch siete.

- **Autonómnosť:** Prvou požiadavkou na detekčný systém je minimálna intervencia človeka. Systém by mal jednať autonómne a to v zmysle detekčných metód, ktoré by nemali byť definované ľudskou zložkou, ale mali by byť plne automatizované. Táto automatizácia je možná v prípade existencie arbitra – učiteľa, ktorý disponuje expertnou znalosťou o incidentoch, ktoré vstupujú do detekčných metód. Výstupom

systému je ohodnotenie analyzovaných dát ako odpoveď na otázku, či sa jedná o útok alebo o validné správanie, ohodnotenie nemusí byť exaktné, ale môže byť reprezentované neurčitou hodnotou (napr. fuzzy alebo pravdepodobnosťou). Systém by ale nemal autonómne reagovať na prebiehajúci útok za účelom jeho zastavenia.

- **Nezávislosť:** Navrhnutý systém by nemal byť závislý na technologických aspektoch analyzovaných dát, type siete, operačných systémoch, použitých technológií, zariadení, protokolov apod. Dôraz je rovnako kladený na nezávislosť na tom, či je v rámci analyzovanej siete použité šifrovanie. Ďalším dôležitým faktorom je nezávislosť na báze znalostí. Predpokladá sa, že detekčný systém nasadený do nového prostredia nie je prispôsobený a nemá znalosť o detailoch ani neprináša znalosti o útokoch špecifických pre dané prostredie.

- **Obmedzenia detekčného systému**

  - Pre detekčný systém budú v rámci tejto práce použité len protokoly/architektúra TCP/IP a to z dôvodu zámerného vypustenia UDP protokolu pre zjednodušenie vytvárania a návrhu analýzy spojení (pri UDP spojení nie je jednoznačne určiteľný začiatok a koniec komunikácie). Problematika UDP komunikácií nie je súčasťou tejto práce.
  - Jednou z požiadaviek na detekčný systém je konštantná dĺžka vytvorených signatúr a to z dôvodu rôznych obmedzení detekčných metód (napr. vstup do klasifikačných metód). Toto obmedzenie platí i pre použitie metrík, ktorých veľkosť výstupu nesmie byť závislá na vstupe (napr. na veľkosti alebo počte paketov v spojení), ale každá metrika[10] musí mať konštantne veľký výstup.

Ďalším cieľom je zapojenie systémov honeypot do procesu získavania informácií o aktuálne prebiehajúcich útokoch a využitie týchto informácií ako expertnej znalosti pri učení klasifikačných algoritmov k detekcii identifikovaných útokov zo sieťového toku. Vzhľadom na zložitosti procesu určenia podmienok kladených na vstupné dáta do učiaceho algoritmu, je potrebné vytvoriť matematický model minimalizovaného sieťového protokolu a následne sady metrík pre popis sieťových spojení. Výstup zo sady metrík by mal tvoriť signatúru sieťového spojenia, ktorá bude využitá ako vstupný vektor pri učení identifikovaných útokov i následne pri analýze sieťového toku. Hlavnou požiadavkou na formát signatúry sieťového spojenia je jej konštantná dĺžka pre všetky analyzované komunikácie, ktorá je dôležitá pre rad klasifikačných algoritmov. Signatúra by ďalej mala čo najvernejšie sprostredkovať behaviorálne aspekty komunikácie modelovaním správania účastníkov komunikácie využitím atribútov sieťového toku. Predpokladom je vysoká účinnosť detekcie sieťových útokov typu buffer overflow pomocou koeficientov aproximačných funkcií priebehu spojenia.

Prípady útokov uvedených v predchádzajúcich kapitolách a aktuálna situácia v oblasti bezpečnosti na Internete vytvára podmienky pre vznik rôznych bezpečnostných technológií. Pokročilé útoky, *APT* a sofistikovaný malware vyžadujú sofistikované riešenia obrany. Vzhľadom na diverzitu zariadení pripojených do Internetu, ktorú prinášajú moderné smery ako *Internet of Things*, je potrebné zamerať sa na bezpečnostné technológie, ktoré nie sú

[10]Metrikou sa rozumie funkcia, ktorej výstupom je číslo alebo definovaná postupnosť čísiel konštantnej dĺžky. Pojem *metrika* v tejto práci nemá nič spoločné s funkciou *metrického priestoru* ako matematickej štruktúry.

závislé na technológii monitorovaných zariadení a ktoré prinášajú zvýšenie bezpečnosti nielen na Internete, ale v akejkoľvek sieti. Motivácia a prínos práce tak tkvejú v komplexnosti, nezávislosti a autonómnosti navrhnutého systému, ktorý adresuje moderné formy útokov a zameriava sa na ich detekciu.

1.2 Prínosy práce

Cieľom tejto práce je návrh autonómneho detekčného systému so zameraním na behaviorálnu analýzu sieťového toku a rozpoznanie útokov na základe anomálií.

Definícia modelu detekčného systému. Vo vedeckej literatúre je možné nájsť rôzne systémy a nástroje, ktoré majú rovnaký alebo podobný účel – zbierajú sieťový tok a pomocou definovaných metrík alebo pravidiel vytvárajú signatúry (zložené z hodnôt, často nazývanými rysy, z angl. *features*), ktoré sú použité ako vstup do klasifikačných algoritmov alebo metód strojového učenia (pozri kapitolu 2). Tieto práce ale nedefinujú exaktný postup pri vytvorení metrík, či signatúr, ale tieto funkcie sú často autormi vymýšľané, prípadne doplňované z iných prác. Takto je možné pozorovať prístupy, kde chýbajú základné informácie sieťového toku, metriky sú neúplné alebo množina metrík je zmes funkcií a pravidiel, ktoré nie sú štrukturálne a logické oddelené na abstraktnej úrovni (sú použité základné štatistické funkcie nad parametrami paketov a zároveň logické funkcie pre modelovanie procesu komunikácie). Jedným z prínosov tejto práce je ucelený prístup k definícii modelu detekčného systému, charakteristiky spojenia, metrík, signatúry spojenia (pozri kapitolu 3), až po samotné detekčné metódy s dôrazom na exaktnosť v definícii postupu a zároveň rozšíriteľnosti systému o ďalšie funkcie a metódy.

Vytvorenie metrík popisujúcich správanie. Vo výskumných prácach je ďalej možné pozorovať použitie základných štatistických metrík, kde detekcia je daná porovnávaním odchyliek od štatistického priemeru, prípadne zhlukovaním štatistických atribútov alebo vyhľadávanie anomálií na základe pravdepodobnostnej príslušnosti do definovaných tried. Tieto základné metriky uvádzané v literatúre (pozri kapitolu 2) dokážu dosiahnuť dobré výsledky v detekcii základných útokov (hlavne útokov, ktoré spôsobujú značné odchýlky v štatistických parametroch toku, ako napr. počet paketov v časovom intervale pri DoS a brute-force útokoch, únik dát odosielaním veľkých tokov na neznáme adresy atď.), ale zlyhávajú pri ďaleko jednoduchších útokoch, ktoré nemajú volumetrický charakter (napr. malware, ktorý komunikuje v pravidelných intervaloch na server útočníka) a komplexnejších alebo zložitejších útokoch (napr. buffer overflow útok na službu so spustením príkazového riadku namiesto práce s dátami v rámci pôvodnej funkcionality služby). Tento problém je adresovaný dvoma prístupmi:

- Vytvorenie zložitejších metrík, ktoré dokážu pridať do signatúry spojenia parametre popisujúce správanie/priebeh danej komunikácie.
- Modulárny návrh detekčného jadra systému, ktorý (v prípade, že má prostriedky a komunikácia je základnými metódami ohodnotená ako podozrivá), je automaticky použitá ďalšia vrstva metód pre upresnenie ohodnotenia.

Vytvorenie kontextu spojenia. Pre lepšiu detekčnú schopnosť systému je potrebné poznať okolie prebiehajúcej komunikácie. Niektoré útoky vytvárajú zvlášť spojenia, ktoré

v prípade analýzy izolovaných spojení nie je možné poznať. Pomocou histórie kontextu je možné napríklad poznať, či dané koncové body spolu komunikovali a ako táto komunikácia prebiehala a je možné poznať, či ide napríklad o pravidelnú komunikáciu príznačnú pre malware alebo hľadanie vzorov v časovom rozložení kontextu daného systému/užívateľa. Kontextom spojenia je ďalej možné vytvárať väzby a komunikačné cesty medzi jednotlivými prvkami a systémami pre detekciu nadväznosti udalostí, pozorovanie vzorov a ich klasifikácia.

Autonómnosť systému. Detekčný systém by spravidla nemal vyžadovať veľkú intervenciu človeka a mal by byť schopný, do určitej miery, učiť sa a v čase rozširovať znalostnú bázu na základe vstupov zo siete/systémov. Detekčné systémy bez učiteľa majú nevýhodu vo vysokom počte falošných detekcií (detekcia útoku v prípade, že ide o validnú komunikáciu) a to z dôvodu, že nepoznajú na 100 %, že ide o útok, prípadne nie sú schopné spätnej väzby. Na druhej strane, učenie s učiteľom závisí na expertnej znalosti dodávaných informácií a zväčša potrebu zásahu človeka. Tento problém je riešiteľný pomocou honeypot systémov, ktoré v prípade, že sú navrhnuté tak, aby dodávali nulové falošné detekcie (napr. dodávali do systému len informácie o útokoch, ktoré sú schopné identifikovať na takmer 100 %) a pridávajú tak autonómnosť systému, ktorý je schopný reagovať bez intervencie človeka na aktuálne prebiehajúce útoky.

Očakávané prínosy práce je tak možné zhrnúť do nasledujúcich bodov:

- Vytvorenie uceleného, jasne definovaného modelu získavania informácií zo sieťového toku, vrátane vytvorenia exaktného postupu definície metrík a signatúry.
- Návrh metrík popisujúcich priebeh spojenia tak, aby reflektovali určité charakteristiky správania, ale zároveň spĺňali obmedzujúce požiadavky na metriky (napr. konštantne veľký výstup).
- Kontext spojenia, ktorým je možné modelovať väzby medzi sieťovými prvkami, časovú postupnosť spojení a pozorovať tak vzory správania analyzovaného systému alebo spojenia.
- Doplnenie detekčného systému o autonómnosť pomocou honeypot systémov, ktoré dodávajú systému expertnú znalosť o prebiehajúcich útokoch na infraštruktúru (monitorovanú alebo externú).

Uvedené prínosy práce by mali viesť k lepším výsledkom detekcie, mali by priniesť širší záber v možnosti detekcie rôznych foriem útokov, ale aj možnosť veľkej variability v pridávaní vlastných častí detekčného systému a nadstavieb metód detekcie. V ďalších kapitolách sú uvedené základne poznatky z bezpečnostných technológií, ktoré sú použité v nasledujúcom jadre práce.

1.3 Definícia použitých pojmov

Majme množinu vzoriek, ktoré budú v tejto práci reprezentované dátami získanými zo sieťového toku. V prípade, že ide o sieťový útok, budeme hovoriť o pozitívnej vzorke, tzn. o vzorke, ktorá reprezentuje útok. Pokiaľ nejde o útok, ale o validnú komunikáciu, budeme hovoriť o negatívnej vzorke. U klasifikácii (ako i v iných oblastiach) sú zaužívané

termíny *true positive* (TP), *false positive* (FP), *true negative* (TN) a *false negative* (FN). Pre ilustráciu:

- *true positive (TP)* značí správne identifikovaný útok;
- *false positive (FP)* značí nesprávne identifikovaný útok;
- *true negative (TN)* značí správne identifikovanú validnú komunikáciu;
- *false negative (FN)* značí nesprávne identifikovanú validnú komunikáciu.

Senzitivita klasifikácie (z angl. *sensitivity*, označovaná aj ako *recall, hit rate* alebo *true positive rate, TPR*) – podiel správne identifikovaných pozitívnych vzoriek voči všetkým pozitívnym vzorkám, senzitivitu R je možné zapísať ako:

$$R = \frac{TP}{TP + FN}.$$

Špecifickosť klasifikácie (z angl. *specificity*, býva označovaná ako *true negative rate*) je podiel správne identifikovaných negatívnych vzoriek voči všetkým negatívnym vzorkám. Špecifickosť S je označovaná ako:

$$S = \frac{TN}{TN + FP}.$$

V praxi sú tieto dve veličiny previazané, často zvyšovanie jednej znižuje druhú.

Precíznosť klasifikácie (z angl. *precision* alebo *positive predictive value*, prípadne *negative predictive value* pre negatívne vzorky) – podiel správne identifikovaných pozitívnych vzoriek voči všetkým pozitívnym vzorkám. Precíznosť PPV a NPV je určená vzťahom:

$$PPV = \frac{TP}{TP + FP}.$$

$$NPV = \frac{TN}{TN + FN}.$$

Účinnosť sa vzťahuje na správne identifikované pozitívne vzorky ako precíznosť, ale účinnosť je pomer správne identifikovaných pozitívnych vzoriek, k celkovému počtu správne identifikovaných (ako pozitívnych tak i negatívnych) vzoriek. Účinnosť ale lepšie zobrazuje trend nesprávne identifikovaných pozitívnych vzoriek, keď sa pohybuje v nízkych hodnotách [44]. Účinnosť U je definovaná ako:

$$U = \frac{TP}{TP + TN}$$

Presnosť (z angl. Accuracy) je pomer správne identifikovaných vzoriek voči všetkým vzorkám. Precíznosť A je možné určiť na základe uvedeného vzťahu:

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

Vzhľadom na to, že uvedené štatistické výpočty (senzitivita, špecifickosť, precíznosť, účinnosť a presnosť) môžu byť náchylné k skresleniu štatistických dát, hlavne pri použití nerovnomerných vzoriek (napr. nepomer vzoriek útokov k validnej komunikácii), budú v rámci práce uvádzané i tzv. hodnoty F-score [119, 55]:

$$F = 2 \times \frac{U \times R}{U + R}$$

Uvedené štatistické funkcie sa bežne používajú pri meraní účinnosti bezpečnostných metód a techník. V nasledujúcich kapitolách budú uvedené rôzne bezpečnostné technológie, ktoré podľa literatúry dosahujú rôzne výsledky, ktoré sú často uvádzané bez odkazu na použitú štatistickú metódu alebo povahu vstupných hodnôt použitých pri výpočte. Väčšina z nich výsledky neuvádza, prípadne sú výsledky skreslené vstupnou množinou dát (rôzne úpravy, rôzne použité databázy apod.) a čitateľ by mal byť upozornený, že porovnanie týchto výsledkov je skreslené (pre viac o výsledkoch porovnávaných metód pozri kapitolu 2.3). Pred uvedením jednotlivých metód a nástrojov je ale potrebné predstaviť základné informácie z oblasti bezpečnostných technológií.

1.4 Bezpečnostné technológie

Medzi základné bezpečnostné technológie už z počiatku vzniku malware patria antivírusy (ďalej len AV), ktoré sú buď nasadzované na koncové stanice, ktoré chránia pred malware alebo sú využívané pre kontrolu súborov v ceste ku koncovej stanici (napr. kontrola príloh emailov na emailovom serveri, alebo webovej proxy pre sťahované súbory apod.). Základnou funkcionalitou AV je vyhľadávanie známych vzorov (tzv. signatúr) v analyzovaných súboroch. Tieto signatúry sú bytové reťazce identifikované v rámci zachyteného malware. Časom k funkcionalitám AV technológií pribudla analýza správania malware, pri ktorej sa zaznamenáva činnosť procesu a v prípade vykonania neštandardných operácií je zvyčajne užívateľ upozornený na podozrivú aktivitu. Ďalej AV technológie začali implementovať personálny firewall, ktorý nahrádzal vtedy nedostatočné implementácie firewallu na operačnom systéme Microsoft Windows a postupom času a rozrastaním webových a emailových technológií, i rôzne analýzy bezpečnosti webových stránok, SPAM filtre, moduly pre analýzu príloh emailov, dokonca i sandbox technológie pre izolovanie niektorých rizikových procesov od niektorých prostriedkov operačného systému. Antivírusy sa od začiatku sústredili hlavne na ochranu koncového užívateľa a ochrana zabezpečenia voči útokom zo siete bola ponechaná na technológiu sieťových firewallov (ďalej len FW). Technológia FW slúži na filtrovanie sieťového toku na koncové stanice podľa definovaných pravidiel. Základnou filozofiou tejto technológie je na sieťovej vrstve zakázať všetku komunikáciu na chránené podsiete a len v prípade explicitného pravidla povoliť komunikáciu z povolených zariadení alebo podsietí na definované služby konkrétnych zariadení podľa daného pravidla. I táto technológia sa postupom času pretvárala a poskytovala stále viac funkcionalít pre lepší management pravidiel, vznikali zónové firewally, pribúdali funkcie na management vyšších vrstiev (až po aplikačnú) a granularita pravidiel na povolenie prístupov sa presunula z povolenia zariadení na konkrétnych užívateľov. Technológia firewallov postupne začala preberať funkcionalitu systémov pre detekciu narušení (z angl. *Intrusion Detection Systems*, ďalej len IDS).

1.4.1 Systémy pre detekciu prieniku (IDS)

Myšlienky obrany voči útokom a prvé návrhy architektúr IDS systémov siahajú až do roku 1985 [44] a s nastupujúcimi technológiami paketových sietí sa presúvajú tieto myšlienky do sieťových bezpečnostných technológií. Sieťové IDS systémy sú primárne určené na monitorovanie udalostí na počítačovom systéme alebo sieti a ich analýzu voči možným incidentom, ktorými sú porušenie alebo bezprostredná hrozba porušenia bezpečnostných politík [116]. IPS systémy (z angl. Intrusion Prevention Systems) sú systémy, ktoré majú všetky funkcionality IDS systémov a naviac sa snažia zastaviť potenciálne incidenty. V prípade, že sa jedná obecne o tieto technológie bez rozlíšenia IDS a IPS, bývajú označované ako IDPS technológie (z angl. *Intrusion Detection and Prevention Systems*). Vzhľadom na zameranie tejto práce bude v texte odkazované iba na IDS technológie, kde oblasť prevencie voči potenciálnym incidentom je nad rámec tejto práce.

IDS technológie sa delia na štyri skupiny podľa zamerania na určitý typ udalostí a incidentov a zároveň podľa toho ako sú tieto technológie nasadené [116]:

- **Hostiteľské** (HIDS) – monitorujú charakteristiku a udalosti v rámci jedného počítača pre identifikáciu podozrivých činností. Zameriavajú sa na analýzu systémových udalostí, procesov, konfigurácií a sieťového toku daného počítača.
- **Sieťové** (NIDS) – monitorujú tok na sieti a analyzujú sieťový a aplikačný protokol pre identifikáciu podozrivých aktivít.
- **Bezdrôtové** (WIDS) – monitorujú aktivity v bezdrôtových sieťach v rámci bezdrôtových protokolov, nedokážu detegovať incidenty vo vyšších protokoloch, ktoré sú zapuzdrené v bezdrôtovom protokole.
- **Sieťová behaviorálna analýza** (NBA) – monitoruje podobne ako sieťové IDS tok na sieti, ale je primárne určená pre analýzu štatistických parametrov tokov a zameriava sa na udalosti, ktoré súvisia s neštandardnými parametrami zachytených komunikácií.

Väčšina uvedených typov IDS technológií používa kombinácie detekčných techník a metód na zvýšenie miery detekcie rôznych typov útokov. Základné typy detekčných metód je možné rozdeliť do troch tried [116]:

- **Metódy založené na signatúre** – porovnávajú signatúry z bázy znalostí s analyzovanými udalosťami a dátami v sieti pre identifikáciu incidentov. Tieto metódy sú efektívne pri detekcii známych útokov a vopred identifikovaných konkrétnych vzoriek malware, ale nedokážu detegovať neznáme hrozby alebo variácie známych hrozieb.
- **Metódy založené na anomáliách** – porovnávajú definované charakteristiky validného správania s pozorovanými udalosťami a na základe identifikácie charakteristík, ktoré sa príliš odkláňajú od naučenej množiny validných vzoriek identifikujú možné incidenty. Tieto metódy môžu dosahovať dobré výsledky pri detekcii neznámych hrozieb, ale ich hlavnou nevýhodou je vytváranie príliš veľa false-positive udalostí, prípadne vytvorenie definície validného správania pri pozorovaní útoku alebo nedostatočne komplexné vyjadrenie charakteristiky, ktoré nedokáže pokryť diverzitu validných udalostí.
- **Metódy založené na analýze protokolov** – modelujú známe sieťové protokoly pomocou definícií charakteristík týchto protokolov. Detekcia incidentov prebieha porovnávaním charakteristík analyzovaného správania s definíciou čo je pre daný protokol

validné správanie a čo nie je. Tieto metódy majú schopnosť modelovať protokoly do takej miery, že dokážu odhaliť útoky, kde iné typy metód zlyhajú, ale bývajú výpočtovo veľmi náročné majú problémy pri útokoch, ktoré nevykazujú anomálie z normálneho správania daného protokolu a je náročné obsiahnuť všetky stavy sieťových protokolov pre podrobnú analýzu.

Vo vedeckých prácach, ale i v komerčných produktoch, sú skúmané a implementované rôzne variácie uvedených typov metód, hybridné IDS systémy, ktoré vznikajú za účelom zvýšenia efektivity detekcie bezpečnostných incidentov. Ich prehľad so stručným popisom a uvedenej úspešnosti je možné nájsť v kapitole 2.

1.4.2 Honeypoty

Ďalšou technológiou, ktorá sa odkláňa od uvedených smerov, ale je zároveň dôležitá pre účely tejto práce, sú honeypot systémy. Honeypot je systém, ktorý vykonáva funkciu návnady pre potencionálneho útočníka alebo malware (odtiaľ pomenovanie honeypot – medový hrniec). Tieto systémy bývajú často úmyselne zraniteľné s neaktualizovaným softwarovým vybavením. Navonok v sieti tak pôsobia ako vhodný cieľ pre potenciálny útok, vlastný operačný systém (prípadne virtualizačná technológia) je ale pozmenený tak, aby v čo najväčšej miere zachytil samotný útok pre jeho ďalšiu analýzu. Dôležitým faktorom u honeypot systémov je ich nerozoznateľnosť od bežného systému a to ako z pohľadu sieťových služieb tak i lokálneho systému[11]. Táto požiadavka je dôležitá, pretože ak malware alebo útočník dokážu rozpoznať, že ide o nastražený systém, ich správanie sa mení a stávajú sa neaktívnymi, prípadne honeypot systém opustia. Tieto systémy sa používajú hlavne vo výskumnom prostredí akademických alebo i komerčných spoločností (napr. antivírusové spoločnosti), ktorí sa snažia analyzovať potencionálne útoky a hľadať nové vektory útokov, nové zraniteľnosti, nové druhy malware alebo nové postupy útočníkov. V rámci výskumu sa honeypot systémy taktiež používajú pri získavaní znalostí pre IDS systémy, konkrétne ako vstup pre klasifikačné algoritmy pre detekciu útokov zachytených na honeypot systéme. Honeypoty sa delia podľa miery interakcie s útočníkom alebo malware [101]:

- **Honeypoty s nízkou mierou interakcie** – tieto honeypoty poskytujú emulované služby, ktoré majú obmedzenú funkcionalitu a zväčša nie sú implementované ani všetky funkcionality emulovanej služby. Priorita týchto systémov je identifikovať pokus o útok a jeho zastavenie pomocou blokácie IP adresy útočníka v rámci celej chránenej siete.
- **Honeypoty s vysokou mierou interakcie** – sú schopné útočníkovi dovoliť interakciu s celým operačným systémom tak, akoby sa nejednalo o honeypot. Účelom takýchto systémov je čo najväčšia miera detekcie a analýzy správania útočníka na systéme. Tieto systémy sú ale komplikovanejšie a náročnejšie na zdroje. Veľkou nevýhodou týchto systémov je možnosť útočníka použiť zdroje tohoto systému k vedeniu ďalších útokov v rámci siete.

V literatúre [101] sa uvádza rozdelenie i na honeypot so strednou mierou interakcie, ktorý emuluje služby i OS vo väčšej miere a mierou interakcie sa nachádza niekde medzi

[11] Na tému detekcie honeypot systému bola pod vedením autora tejto práce vypracovaná bakalárska práca "Detekce honeypot systémů v síti" [126]

dvoma uvedenými typmi. Špeciálnym prípadom je tzv. *Shadow honeypot* [6], ktorý sa používa v spojení so systémami pre detekciu anomálií (tzv. *Anomaly Detection Systems*, skr. ADS). V prípade, že ADS zachytí komunikáciu, ktorá je podozrivá (vykazuje anomálie voči naučenému modelu správania), je tento tok poslaný na Shadow honeypot systém, ktorý zastáva funkciu pôvodného systému, akurát požiadavky nevykonáva nad reálnymi dátami, prípadne sú zmeny dát vrátené do pôvodnej podoby. Okrem využitia honeypot systémov na detekciu útokov a ich analýzu za účelom zistenia a pochopenia nových vektorov útokov, nájdenie nových zraniteľností a malware, sa časť výskumu sústredí na použitie honeypot systémov na automatické generovanie signatúr pre klasifikáciu sieťového toku. Obecná klasifikácia sieťového toku je posledných 10 – 15 rokov významnou súčasťou vedeckých štúdií a prác z oblasti bezpečnosti a posledných niekoľko rokov vznikajú i rôzne komerčné nástroje, ktoré využívajú poznatky z klasifikácie sieťového toku pre účely zvýšenia bezpečnosti. Klasifikáciu toku je možné rozdeliť do niekoľkých obecných kategórií v závislosti na pohľade na uvedený problém.

V prípade potreby identifikácie toku za účelom rozpoznania aplikačnej vrstvy je hlavným motivačným faktorom poznanie aké aplikácie a následne koncové body vytvárajú tento tok v sieti. Príkladom môže byť potreba identifikácie P2P (peer-to-peer) sietí z dôvodu manažmentu priepustnosti a spoľahlivosti v analyzovanej sieti [80]. Hlavným dôvodom je fakt, že P2P siete prenášajú veľké množstvo dát a vyťažujú linku na úkor ostatných užívateľov a tak môže byť identifikácia z pohľadu poskytovateľa linky a operátora a následné obmedzenie tejto komunikácie dôležitým faktorom pri udržovaní kvality služieb. Tento príklad sa vzťahuje na rôzne aplikácie a protokoly, ktoré je možné identifikovať v sieťovom toku za účelom ich identifikácie, prípadne klasifikácie do tried (VoIP, P2P, HTTPS, apod.).

Z pohľadu bezpečnosti sa v obecnej rovine kladie za cieľ rozpoznanie útokov v sieťovom toku v danom uzle siete, prípadne týmto útokom po ich identifikácii v čo najkratšom, ideálne reálnom, čase zabrániť. Útoky, ktoré môžu byť identifikované z dát sieťového toku je možné rozdeliť na tri hlavné skupiny:

- **Lokálny útok** – útok, ktorý prebehol na koncovej stanici užívateľa, identifikácia v rámci siete je komplikovaná (vzhľadom na fakt, že samotný útok je vykonaný lokálne na počítači obete), je možná napr. pri doručovaní malware na koncovú stanicu alebo po prejavení malware (komunikácia smerom z infikovanej stanice).
- **Sieťový útok na aplikačnej vrstve** – napr. exploitáciou sieťových služieb alebo aplikácií. V rámci toku je identifikovaná časť samotného exploitu, ktorý tak, ako v predchádzajúcom prípade, musel byť v minulosti analyzovaný (a vytvorená jeho signatúra) alebo ide o útoky na aplikačnej vrstve (napr. webové aplikácie a služby).
- **Útoky na sieťovej vrstve** – môže ísť z pohľadu obsahu o validnú komunikáciu, ktorá ale môže byť vedená ako útok na sieťovej vrstve (napr. pokus o zahltenie služby/systému, ukradnutie identity stanice a pod.).

Pri klasifikácii uvedených útokov je možné použiť dva hlavné prístupy. V prvom prípade ide o analýzu obsahu dátovej časti komunikácie[12]. Táto forma analýzy je vhodná len v prípade, že komunikácia nie je šifrovaná a ide o známy protokol, ktorý je analyzovateľný vyhľadávaním vzorov v rámci dátovej časti komunikácie. V druhom prípade ide o analýzu sieťového toku bez analýzy dátovej časti a to rôznymi metódami (popísané v rámci kapitoly 2), ktoré modelujú sieťové spojenia alebo prvky a vyhľadávajú v toku anomálie prípadne

[12]Tento prístup v paketových sieťach sa obecne nazýva *Deep Packet Inspection*, skr. DPI.

vytvárajú štatistické modely komunikácie. Týmto prístupom je možné analyzovať dátový tok na prvku v sieti medzi dvoma komunikujúcimi bodmi a s určitou pravdepodobnosťou odhaliť prebiehajúce útoky bez znalosti vlastného obsahu komunikácie.

V prípade lokálneho útoku by sieťová detekčná technológia nedokázala detegovať samotný útok, ale je možné predpokladať, že pri vydarenom útoku začne útočník alebo malware z infikovaného stroja pristupovať do Internetu (napr. malware, ktorý pravidelne zisťuje príkazy pre svoju ďalšiu činnosť) alebo k ďalším systémom v rámci internej siete pre šírenie infekcie. Pri útokoch na aplikačnej vrstve sú efektívne prístupy analýzy aplikačného protokolu a vyhľadávanie signatúr v sieťovom toku.

Ďalšie sieťové bezpečnostné technológie, ktoré neboli uvedené v tomto texte nesúvisia priamo so zameraním práce a nebudú v práci popisované. Historický vývoj a aktuálne trendy vo vývoji sieťových bezpečnostných technológií na báze IDS a NBA (sieťová behaviorálna analýza, z angl. *Network Behavioral Analysis*) sú popísané v kapitole 2.

1.5 Štruktúra práce

Kapitola 2 sa zaoberá aktuálnym stavom bádania (tzv. *State of the Art*) v detekčných metódach sieťových útokov. V kapitole sú stručne popísané jednotlivé technológie detekcie útokov zo sieťového toku, prehľad honeypot systémov a v závere je uvedený chronologický prehľad vo vývoji a zhodnotenie účinnosti sieťových detekčných systémov.

Kapitola 3 definuje minimalistický sieťový protokol nad ktorým je vytvorený formálny systém charakterizujúci sieťové spojenia a sústava metrík, ktorých výstupom je signatúra analyzovaného spojenia. Časť kapitoly je venovaná rozšíreniu minimalistického protokolu na TCP/IPv4 architektúru a prispôsobenie navrhnutého systému jej protokolom a parametrom sieťového toku. V závere kapitoly je uvedená definícia kontextu spojenia a sú načrtnuté metódy a algoritmy analýzy navrhnutých signatúr so zameraním na klasifikáciu tokov na útoky a validnú komunikáciu.

Kapitola 4 popisuje celkovú architektúru navrhnutého systému, procesy v rámci systému, jeho vstupy, výstupy a obmedzenia. Popis architektúry sa zameriava na analýzu sieťových tokov v reálnom čase a ich klasifikáciu s cieľom identifikácie sieťových útokov. V rámci architektúry sú načrtnuté možnosti systému v analýze útokov a možné rozšírenia o ďalšie vrstvy analýzy. Kapitola je ukončená popisom modelu detekčného systému.

Kapitola 5 predkladá čitateľovi popis laboratórneho prostredia v ktorom boli vykonané základné experimenty a simulácie pri návrhu a testovaní uvedeného modelu systému. Výsledky experimentov v laboratórnom prostredí dopĺňajú výsledky z testovania voči existujúcim dátovými sadám útokov a skúsenosti z reálneho nasadenia systému v prostredí univerzitnej siete kampusu VUT v Brně. Experimenty uzatvára štúdia prípadu použitia experimentálnych metód pri detekcii reálneho útoku na vybrané sieťové služby zraniteľného systému.

Kapitola 6 zhodnocuje navrhnutý systém, definované sieťové metriky a použité metódy, porovnáva dosiahnuté výsledky a popisuje využitie systému v reálnom prostredí. V kapitole je ďalej uvedený prínos práce, zhodnotenie dosiahnutých cieľov, budúcnosť projektu a možnosti jeho prípadného rozšírenia.

Kapitola 2

Aktuálny stav bádania v detekčných metódach sieťových útokov

Podľa AV-Test inštitútu[1] sú čísla, ktoré reprezentujú počet unikátnych vzoriek malware alarmujúce. Za rok 2012 bolo identifikovaných 38 mil. nových vzoriek, v roku 2013 prekročila táto hodnota hranicu 80 mil. vzoriek a v roku 2014 bola prekonaná hranica 140 miliónov. Podľa prehľadovej bezpečnostnej správy od spoločnosti Kaspersky [85] za rok 2014 bolo ich systémami na Internete detegovaných viac ako 123 miliónov unikátnych malígnych (z angl. *malicious*) objektov. Vzhľadom na fakt, že sa každoročne počet unikátnych signatúr pre identifikovaný malware násobí, nie je možné naďalej udržiavať metódy, ktoré sú založené na porovnávaní signatúr[2] malware (napr. pomocou antivírusových technológií na staniciach užívateľov). Z tohto dôvodu sa za posledných 20 rokov (analyzované štúdie a výskumné práce medzi rokmi 1995 a 2015) sústreďuje pozornosť vedeckých prác na detekciu sieťových útokov pomocou signatúr, ktoré nie sú závislé na obsahu – dátovej časti paketov (tzv. *packet payload*) a vznikajú nové projekty na automatizovanú detekciu na základe detekcie anomálií v sieťovom toku. Pre vylepšenie existujúcich detekčných metód vznikali štúdie, ktoré sa zaoberajú automatizovaným generovaním signatúr pre zvýšenie efektivity týchto metód a nástrojov, ktoré tieto metódy využívajú.

Metódy na detekciu malware je možné všeobecne rozdeliť do dvoch hlavných kategórií [67] (pre viac informácií pozri obrázok 2.1):

- **Metódy založené na signatúrach** – vytvárajú signatúry (popis útoku, časť dát paketu, časť programu apod.) a porovnaním tejto signatúry detegujú útoky, prípadne malware.
- **Metódy založené na detekcii anomálií** – požívajú znalosti, ktoré definujú, čo je normálny stav systému, pre rozhodnutie, či ide o útok alebo nie. Toto rozdelenie bolo uvedené v článku [67], kde bola uvedená i podoblasť metód založených na anomáliách a to metódy založené na špecifikácii. Tieto metódy obsahujú preddefinované pravidlá (špecifikáciu), ktoré popisujú validné správanie a odklon od tohoto správania je označený ako útok.

[1] Aktuálny prehľad štatistík je možné nájsť na adrese `https://www.av-test.org/en/statistics/malware/`

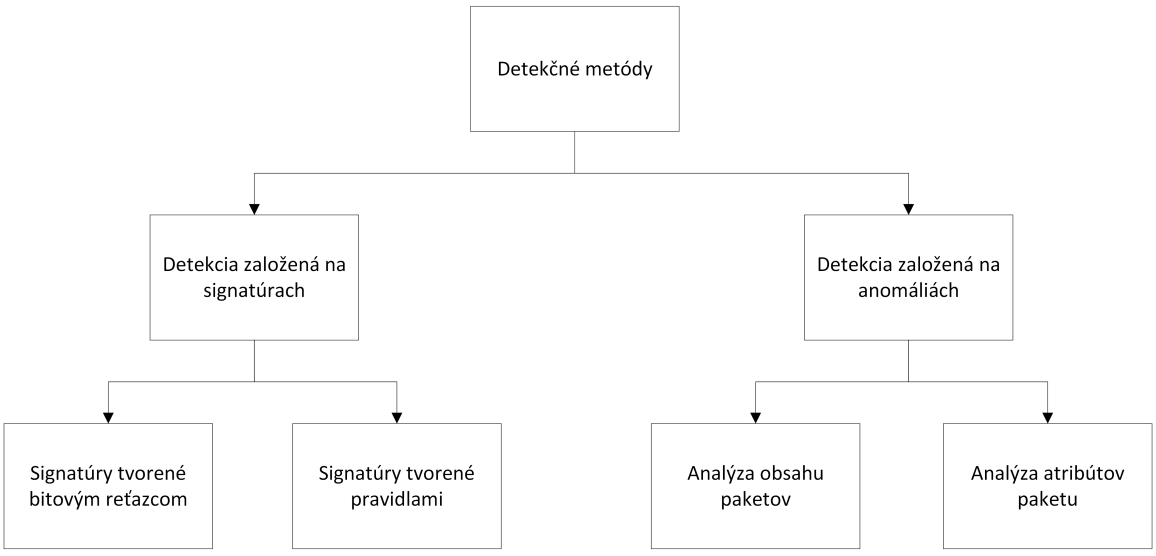
[2] Signatúry sú vzory, ktoré sú vyhľadávané v rámci analyzovaných dát (napr. dátach sieťového toku alebo súboroch v rámci operačného systému) za účelom identifikácie malware

Tento popis metód bol primárne určený pre rozdelenie prístupov ku klasifikácii programov a súborov, ktoré sú analyzované detekčnými systémami v prostredí operačného systému. S tým súvisí i ďalšie rozdelenie metód na analýzu programov a to na statickú, dynamickú a hybridnú. Statická analýza bez spustenia daného kódu analyzuje syntaktickú a štrukturálnu stránku programu a na základe zistených informácií určuje, či štruktúra programu odpovedá malware. Dynamická analýza skúma daný program počas jeho behu v rámci operačného systému a na základe prejavov správania (analyzuje sa procesor, pamäť, chod programu, stavy programu a pod.) sa určuje či ide o malware alebo validný program. Vzhľadom na zameranie tejto práce na detekciu anomálií zo sieťového toku, nebudú tieto metódy predmetom ďalšieho skúmania, ale pre zaujímavosť čitateľa je možné viac informácií vyhľadať v uvedenej literatúre [30]. V ďalšom texte sa bude práca sústreďovať len na metódy a nástroje určené na detekciu útokov (prípadne malware) v rámci sieťového toku.

2.1 Metódy a nástroje založené na signatúrach

Existujúce metódy založené na signatúrach je možné rozdeliť do dvoch kategórií podľa charakteru vstupných dát týchto metód (ako je vidieť na obr. 2.1):

- **Signatúry založené na obsahu (vzory bytov)** – vstupom týchto metód je dátová časť paketov sieťového toku, v ktorom vyhľadávajú bytové reťazce, ktoré zodpovedajú vzorom v databáze znalostí (signatúr).
- **Signatúry založené na pravidlách** – vstupom sú parametre sieťového toku, ktoré sú porovnávané s predom definovanými pravidlami (signatúrami).



Obr. 2.1: Prehľad rozdelenia detekčných metód.

Príkladom takýchto signatúr je napr. komunikácia z IP adresy, ktorá je na zozname podozrivých adries – ide o signatúru založenú na pravidle a vstupom je parameter sieťového toku (zdrojová IP adresa). Ďalším príkladom je emailová komunikácia, ktorá obsahuje vírus identifikovaný unikátnou postupnosťou bytov – signatúra založená na obsahu. Ako posledný príklad útoku je možné uviesť útok na DNS server pomocou pretečenia pamäte

identifikovaný veľkosťou vstupného reťazca konkrétneho atribútu DNS protokolu – signatúra založená na pravidle i obsahu paketu (kombinácia), ktorej vstupom je predspracovaný obsah paketov.

Oba typy metód sú typické pre bezpečnostné technológie IDS (Intrusion Detection Systems) – systémy na detekciu prienikov[3]. Hlavnými zástupcami IDS vo vedeckej komunite sú open-source nástroje **Snort** [127] a **Bro** [99]. Jednotlivé signatúry útokov sú vytvárané zväčša manuálne a to expertom, ktorý analyzuje komunikáciu a vzorku malware a vytvorí signatúru, ktorá je následne distribuovaná do jednotlivých sieťových IDS zariadení. Tento prístup vykazuje detekciu s relatívne nízkymi hodnotami false-positive, ale náročnosť vytvárania signatúr pre jednotlivé IDS systémy, vzhľadom na zapojenie ľudského faktoru, je príliš komplikovaná. Taktiež vzhľadom na nutnosť manuálnej analýzy malware na infikovanom systéme môže od vzniku malware po distribúciu signatúry na IDS systémy trvať i niekoľko dní.

Hlavnou oblasťou IDS systémov je detekcia známych útokov na základe rozpoznania dopredu známeho reťazca v obsahu sieťovej komunikácie (uvedenej kategórie signatúr založených na obsahu). Identifikácia zraniteľnosti a prejav exploitácie zraniteľnej služby je pre tieto metódy kľúčová. Z každej objavenej a publikovanej zraniteľnosti je pre IDS systémy vytvorená signatúra, ktorá obsahuje časť dát (postupnosť bytov) prenášaných sieťou od útočníka k cieľovému zraniteľnému systému. Detekcia na základe signatúr exploitácie má výborné uplatnenie pri detekcii známych, vopred analyzovaných útokov a to z dôvodu relatívne nízkej hodnoty false-positive a jednoduchej a rýchlej metódy detekcie. Existuje veľa metód založených na rozpoznávaní signatúr exploitov v sieťovom toku, ktoré ale nemajú dobré výsledky v rozpoznávaní útokov, ktorých signatúra nie je obsiahnutá v databáze IDS. Preto boli do IDS systémov zakomponované metódy, ktoré modelujú bežný stav bez útokov a následne vyhľadávajú odchýlky, ktoré tomuto modelu nezodpovedajú a môžu znamenať potenciálny útok [135]. Tieto metódy sú založené na jednoduchých pravidlách (napr. ustanovenie spojenia na definovaný port z jedného zdroja na jeden cieľ v počte väčšom ako stanovený limit za určitý čas, môže byť klasifikované ako útok hrubou silou na službu bežiacu na danom porte). Tieto pravidlá majú výhodu oproti metódam založeným čisto na obsahu hlavne v možnosti definovania pravidiel na sieťové útoky, ktoré nie je možné detegovať pomocou signatúr z dátovej časti paketov, ale prinášajú výrazné zvýšenie false-positive. Vzhľadom na túto nevýhodu sa oblasť výskumu IDS systémov rozšíril na metódy pre automatické vytváranie signatúr na základe rozpoznania útoku priamo na napadnutom systéme a pre tento účel boli vytvorené honeypot systémy (účel bol zameraný na výskum nových zraniteľností a rozpoznanie útoku, vytváranie signatúr bolo skôr manuálne).

2.1.1 Metódy a nástroje na vytváranie signatúr

V prípade štúdií, ktoré sa zameriavajú na vylepšenie tradičných IDS systémov, sa za poslednú dekádu sústreďuje veľa vedeckých prác na automatizáciu generovania signatúr pre tieto systémy. Jedným z hlavných smerov týchto výskumov je automatizácia honeypot systémov, ktoré by dokázali vytvárať signatúry pre rôzne typy útokov a to hlavne bez zásahu a nutnej manuálnej analýzy človekom. Pri automatickom vytváraní signatúr je dôraz kladený hlavne na kvalitu vygenerovaných signatúr. Kvalita signatúr priamo ovplyvňuje detekčné metódy, konkrétne ich presnosť v rozpoznaní (klasifikácii, vyhľadaní) útokov. Signatúry

[3]IDS slúžia na detekciu prienikov do siete, okrem IDS technológií poznáme ešte IPS technológie. Jedná sa o tú istú technológiu, ale v aktívnom móde, to znamená, že je nasadená priamo do cesty medzi prístupovým bodom a cieľovými systémami a dokáže potenciálne útoky blokovať. (P v skratke *IPS* znamená *prevention*).

môžu byť všeobecne rozdelené do dvoch kategórií podľa ich vzniku (procesu generovania):

- **Signatúry založené na exploite** (z angl. exploit-specific) – signatúry založené na exploite sú vytvárané z byte kódu zachyteného exploitu, prípadne dátovej časti paketu, ktorý nesie payload[4] exploitu, čím sú ovplyvnené konkrétnymi vlastnosťami daného analyzovaného exploitu. To môže spôsobiť ich nedostatočnú generickosť pre detekciu rôznych mutácií exploitov určených pre tú istú zraniteľnosť.
- **Signatúry založené na zraniteľnosti** (z angl. vulnerability-driven) – signatúry založené na zraniteľnosti sú vytvorené z vlastností danej zraniteľnosti a nie sú ovplyvnené konkrétnou verziou exploitu, ale tak ako predchádzajúci typ signatúr, bývajú často založené na konkrétnom obsahu dát a nie všetok malware je možné popísať takýmito signatúrami [40, 137].

V článku [73] sa autori zamerali na redukciu false-positive detekčných metód použitím honeypot systému **HoneyD** [107] do ktorého presmerovali komunikáciu v prípade podozrenia na útok a honeypot v prípade rozpoznaného útoku podľa definovaných pravidiel vytvorí signatúru pre IDS systém. Honeypot systémy sú často používané ako expertné systémy pre zníženie false-positive IDS systémov. Vzhľadom na to, že práca sa zameriava hlavne na sieťovú detekciu, samotné honeypot systémy sú len okrajovou časťou navrhnutého systému a nie sú rozoberané do detailu.

Hyang-Ah Kim a Brad Karp v článku [74] sa zamerali na proces vytvárania signatúry pre IDS systémy a hlavne na odstránenie ľudskej zložky z tohoto procesu. Systém **Autograph**, ktorý navrhli, sleduje prichádzajúce spojenia na perimetri siete a generuje signatúry pre sieťové útoky. Konkrétne sa zameriava na červy[5]. Autori využívajú dátovú časť paketov pre automatické vytvorenie signatúry malware, jej distribúciu medzi prepojenými systémami Autograph a následnú detekciu. Vytvorená signatúra sa skladá z bytov vybraných z analyzovaných paketov. Výber bytov je pomocou metódy vyhľadania najčastejšie sa vyskytujúcich postupností bytov. Signatúra má tvar (*dst-port, byteseq*), kde *dst-port* je cieľový port a *byteseq* je vybraná postupnosť bytov. Metóda vytvárania signatúr na základe opakujúcich sa reťazcov bytov bola použitá i v práci Kreibicha a Crowcrofta [75], ktorí popisujú ich systém **Honeycomb**, ktorý sa pri generovaní signatúr pre IDS systémy (podporuje IDS signatúry pre *Snort* a *Bro*) zameriava na honeypot systémy HoneyD [107]. Pre detekciu malware v sieťovom toku nepoužívajú vyhľadávacie algoritmy ako bežné IDS systémy, ale pomocou algoritmu najväčšieho spoločného podreťazca (z angl. *Longest Common Subsequence*, *LCS* [65]) porovnávajú časti sieťového toku (dátovú časť paketov) s časťami komunikácie malware zaznamenaného pomocou systému honeypot. Samotné vytváranie signatúry je postavené iba na prichádzajúcej množine paketov (odchádzajúce pakety nie sú analyzované) s analýzou protokolu na transportnej a sieťovej vrstve. Pre každé zaznamenané spojenie je uchovávaná dátová časť paketov, ktoré sú následne spájané a ukladané pre vytvorenie signatúry. Tá je potom porovnávaná s reálnym tokom na sieti pre vyhľadávanie podozrivej komunikácie. Signatúra je, podobne ako u nástroja *Autograph*, zložená z najčastejšie opakujúcich sa reťazcov bytov. Tieto metódy majú nevýhodu v problematickom zisťovaní, či ide o útok. V prípade false-positive nálezu je vygenerovaná signatúra na validnú komunikáciu, čo propaguje chybu cez IDS systémy a preto sú tieto automatizované metódy doplnené

[4]Payload – týmto pojmom je označovaná časť exploitu, ktorá spustí vykonávanie pripravenej činnosti ako otvorenie prístupu, infikovanie systému; exploit je nosná časť, ktorá umožní spustiť payload.

[5]Červy (z angl. worms) – malware, ktorý sa vyznačuje epidemickým šírením sieťou s vlastným spustiteľným kódom.

o kontrolu človekom – expertom. Na tento problém sa sústreďujú vedecké práce okolo honeypot systémov, ktoré dokážu odhaliť prípadný útok a vzhľadom na ich povahu je každá komunikácia na honeypot systém považovaná za potenciálny útok.

2.1.2 Získavanie informácií z honeypot systémov

„A honeypot is security resource whose value lies in being probed, attacked, or compromised.“

Lance Spitzner
Honeypots: Tracking Hackers, 2002 [120]

Newsome J. a Dawn Song [92] vo svojej práci vytvorili automatizovaný nástroj na analýzu exploitov a generovanie signatúr pre typy zraniteľnosti buffer overflow, **TaintCheck**. Tento nástroj slúži na detekciu útokov, ktoré sa prepísaním ukazovateľa na návratovú adresu, prípadne ofsetu, snažia získať kontrolu nad bežiacim procesom. Metóda detekcie týchto útokov bola pomenovaná **Dynamic Taint Analysis** (ďalej len taint analýza). Metóda sa zakladá na predpoklade, že na spôsobenie samotného pretečenia alokovanej pamäte a uloženie návratovej hodnoty adresy na kód, ktorý vkladá priamo útočník, musí byť návratová hodnota i útočníkov kód súčasťou vstupu od útočníka. Autori sa v taint analýze odkazujú na všetky dáta, ktoré pochádzajú z nedôveryhodných vstupov a môžu tak byť ovplyvnené útočníkom ako poškvrnené (z angl. *tainted*). Ďalej sa sleduje, ako sú tieto dáta propagované programom a použité nebezpečným spôsobom (napr. keď sú použité ako návratová hodnota), čo indikuje prípadnú exploitáciu služby. Taint analýza tak deteguje útoky na zraniteľnosť buffer overflow v dobe ich reálneho vykonania (keď sú dáta skutočne použité neoprávneným spôsobom), čím má metóda takmer 0-vý počet false-positive.

Na Taint analýzu nadviazal projekt **Argos** [104], ktorý princípy *TaintCheck* nástroja aplikoval do virtuálneho prostredia, konkrétne ako modul emulátora na virtualizáciu operačného systému, *QEMU*. *Argos* v rámci emulovaného operačného systému monitoruje fyzický adresový priestor a dáta, ktoré pochádzajú zo siete označuje za poškvrnené. Kedykoľvek sú tieto dáta použité napr. kopírovaním do registru alebo pamäte, nová pozícia je taktiež označená a monitorovaná. V prípade, že tieto označené dáta sú použité pokusom o ich vykonanie (poškvrnené dáta alebo pamäť sú vložené do EIP registru), je detegovaný incident a *Argos* vygeneruje správu s obrazom pamäte, sieťového zásobníka a ďalších informácií. Z týchto dát je následne možné vytvoriť signatúru.

Ďalšie prístupy honeypot systémov v oblasti detekcie útokov a generovania signatúr sa zameriavajú na rôzne smery a využitie týchto systémov. Jedným zo smerov je kombinácia vysoko interaktívnych a nízko interaktívnych honeypotov (tzv. hybridné honeypoty). Zástupcom hybridných honeypotov je napr. nástroj **BitSaucer** [1], ktorý je tvorený honeypotom s nízkou mierou interakcie simulujúcim sieťové služby. Kľúčovou vlastnosťou tohoto honeypotu je proxy služba inštalovaná na bežné (produkčné) počítače v sieti. Táto služba v prípade potreby a podozrenia na útok vytvorí na honeypote virtuálny stroj s emuláciou operačného systému. Ďalšou oblasťou výskumu u honeypot systémov je zabezpečenie šírenia nákazy na systémy v sieti. Alberdi vo svojej práci [3] predstavil monitorovanie aktivity malware na operačnom systéme a v prípade pokusu o šírenie nákazy na sieti sa presmeruje malware na ďalší honeypot. Nízko interaktívny honeypot **Honeyware**[5], ktorý pracuje na strane klientskeho počítača pristupujúceho k webovým stránkam, sa sústreďuje na detekciu web serverov, ktoré sa snažia napadnúť pristupujúcich klientov. Vo vedeckej práci [6] autor Anagnostakis predstavil nový princíp honeypot systémov, tzv. *shadow honeypoty*, ktoré sú

Nástroj	Rok	Typ	Metóda / Cieľ
HoneyD	2003	LIH	Pravidlá na analýzu správania v simulovanej aplikácii
Autograph	2004	Sieťový generátor signatúr	LCS zo sieťového toku
Honeycomb	2004	Generátor založený na LIH	LCS z honeypot systému
TaintCheck	2005	Generátor založený na HIH	Extrakcia paketu s exploitom na honeypot systéme
Argos	2006	HIH	Extrakcia paketu s exploitom na honeypot systéme s pridanými informáciami

Tabuľka 2.1: Prehľad nástrojov na analýzu útokov a automatické vytváranie signatúr (LIH značí Low-Interaction a HIH značí High-Interaction Honeypot).

zakomponované v bežných sieťových aplikáciách. Všetky požiadavky na aplikáciu sú vykonané tak isto ako v prípade normálnej aplikácie, ale príkazy sú monitorované. V prípade, že je detegovaná podozrivá aktivita alebo útok, sú všetky zmeny v rámci aplikačných dát vrátené. Pred vstupom do aplikácie sú všetky požiadavky vyhodnotené analýzou anomálií a v prípade, že je komunikácia označená ako validná, je požiadavka poslaná na produkčný systém, v opačnom prípadne na honeypot. Ďalším zástupcom nových prístupov u honeypot systémov je tzv. *fake honeypot* [114], ktorý ma za úlohu simulovať reálny honeypot, ale bez bežných funkcií honeypotu. Zmysel je odstrašiť prípadných útočníkov, ktorí sa podľa autora vyhýbajú sieťam, v ktorých sú prítomné honeypoty. Ďalšie informácie k honeypot systémom a zaujímavým vedeckým projektom okolo honeypot systémov je možné nájsť v uvedenej literatúre [84, 26, 45].

Za posledných desať rokov boli zraniteľnosti buffer overflow a následne útoky jednou z hlavných metód šírenia malware medzi počítačmi [39]. Podľa štúdie [109] je mesačne publikovaných v priemere 20 až 40 nových zraniteľností v bežne používaných sieťových produktoch. Tieto útoky vo väčšine prípadov, hlavne u zraniteľností v sieťových službách, vedú k získaniu lokálneho účtu na kompromitovanom systéme (pod ktorým beží daná zraniteľná služba, v niektorých prípadoch to môže byť i administrátor). V tabuľke 2.1 je uvedený prehľad niektorých nástrojov, ktoré stáli na počiatku výskumných projektov zameraných na analýzu útokov na napadnutom operačnom systéme s možnosťou automatického generovania signatúr. Vzhľadom na rozšírenie a povahu chýb typu buffer overflow a zameranie tejto práce sa prehľad sústreďuje okolo nástrojov adresujúce túto zraniteľnosť. V ďalšom texte sú uvedené sieťové detekčné nástroje zameriavajúce sa na útoky buffer overflow.

2.1.3 Buffer overflow detekčné metódy a nástroje

Wang, Lanija, a kol. [137] sa vo svojej práci zameriavajú na sieťovú detekciu exploitov zraniteľných služieb pomocou identifikácie a analýzy dátového paketu, ktorý nesie payload exploitu. Ich prístup je obmedzený na oblasť zraniteľností buffer overflow a samotná klasifikácia sa opiera o úsudok, že každý útok pomocou buffer overflow musí byť v určitých atribútoch protokolu dostatočne dlhý, minimálne oveľa dlhší ako pri bežných požiadavkách

na danú službu a tieto myšlienky implementovali do nástroja **LESG** *(Length-based signature generator)*. *LESG* vykonáva analýzu dátovej časti paketov s analýzou aplikačného protokolu v ktorom rozpoznáva jednotlivé PDU[6] *(Protocol Data Units)*, súčasti protokolov, ako napr. DNS alebo HTTP.

V rámci výskumných projektov a prác sa začala okolo projektu Argos vytvárať skupina vedcov, ktorá sa sústredila hlavne na zraniteľnosti buffer overflow a to z dôvodu expertnej znalosti arbitra v podobe systému Argos, ktorý dokáže pomocou rozšírenej taint analýzy detegovať útoky na tieto zraniteľnosti. Aktuálna ochrana proti útokom na zraniteľnosti buffer overflow sa sústreďuje na randomizáciu pamäte a sťaženie pozície útočníka pri vytváraní vlastného kódu exploitu (tzv. *shellcode*) [16, 17, 18]. Tieto ochrany ale nie sú postačujúce a nedokážu dostatočne ochrániť proces, ktorý je napadnutý. Z tohoto dôvodu vznikali ďalšie výskumné práce k automatizácii ochrany proti útokom buffer overflow.

Ďalším projektom, ktorý vznikol pre ochranu systémov a služieb pred týmito útokmi, bol **ARBOR** *(Adaptive Response to Buffer Overflow)* [79]. Tento systém používa už spomenutú metódu (pri nástroji *LESG*), ktorá sa sústredí na vstupné dáta, ktoré v prípade útoku na zraniteľnosť buffer overflow budú v porovnaní s validnou komunikáciou väčšie. Ďalším užitočným kritériom nástroja *ARBOR* je prítomnosť binárnych dát na vstupe. *ARBOR* ďalej pridáva kontext programu z dôvodu rozlíšenia validity binárnych dát na vstupe programu (program môže v rámci autentifikácie očakávať textové dáta, ale po autorizácii môžu byť binárne dáta validným vstupom).

Ďalšie sieťové detekčné metódy útokov buffer overflow sa sústreďujú na detekciu v rámci analyzovaných paketov: **CTCP architektúra** [66] a nástroj **Buttercup** [98] analyzujú sieťový tok v ktorom vyhľadávajú návratové hodnoty po pretečení pamäte. V prípade nástroja *Buttercup* je pre každú zraniteľnosť definovaný interval návratových hodnôt, ktoré sú vyhľadávané v paketoch na rozdiel od *CTCP*, kde sú návratové hodnoty detegované automatizovane. Tento prístup funguje pri známych, dopredu analyzovaných, útokoch, ale pri bežnej komunikácii môže vyvolať veľké množstvo falošných detekcií. Projekt **SHIELD framework** [133], ktorý emuluje časť protokolu na aplikačnej úrovni, vstupuje do procesu predávania dát aplikačnej vrstve, ktoré následne analyzuje. *SHIELD* dokáže odhaliť iba tie útoky, pre ktoré sú dopredu vytvorené politiky, preto má relatívne vysokú precíznosť, ale veľmi pomalú reakciu na novo vytvorené útoky. Tento nástroj spadá do kategórie detekcie na základe signatúr vytvorených podľa danej zraniteľnosti. Nad projektom *SHIELD* vznikol rad ďalších obdobných nástrojov zameraných hlavne na parsovateľné[7] sieťové protokoly ako napr. HTTP.

Existuje veľa ďalších sieťových detekčných metód založených na princípe signatúr alebo na základných štatistických vlastnostiach komunikácie, tieto systémy ale majú veľmi zlú účinnosť pri detekcii neznámych útokov, tzv. *zero-day* útokov alebo útokov ktoré obsahujú nové varianty starších exploitov (ale ktorých signatúra sa zmenila). Potreba detekcie týchto útokov dala vzniknúť novým detekčným metódam založeným na detekcii anomálií.

2.2 Metódy a nástroje založené na detekcii anomálií

Vedecké práce, ktoré sa odklonili od skúmania detekčných metód založených na hľadaní vzorov v sieťovej komunikácii a vytváraní signatúr útokov sa začali viac prikláňať k technikám,

[6] PDU sú reťazce, ktoré sú špecifické pre aplikačný protokol. Napr. u DNS to môže byť *question*, *answer*, *authority* apod., u protokolu FTP to bude napr. *user*, *pass* a u HTTP napr. *GET*, *POST*, *User-Agent*.

[7] Parsovateľný – v zmysle možnosti syntaktického rozboru analyzovaného reťazca.

ktoré analyzujú charakteristiku sieťového toku metódami strojového učenia. Vzhľadom na veľké množstvo prác a výskumov okolo detekcie útokov a IDS systémov vznikla potreba vytvoriť verejnú sadu dát, ktorú je možné použiť pri učení a testovaní detekčných algoritmov, ako aj pri porovnávaní ich úspešnosti.

2.2.1 Dátové sady na testovanie účinnosti sieťových detekčných metód

Za posledných niekoľko rokov sa sústreďuje časť výskumu v oblasti bezpečnosti okolo sieťových detekčných metód na základe ktorých sa identifikujú podozrivé komunikácie. Tieto detekčné metódy využívajú vstupné množiny dát pre naučenie rozpoznávania útokov (klasifikáciu) a preto bolo potrebné vytvoriť dátové sady sieťových záznamov útokov a validných komunikácií, ako pre učenie, tak i testovanie a porovnávanie výsledkov týchto výskumných prác.

1999 KDD Cup – Za účelom vytvorenia dátovej sady zaznamenaných komunikácií pre účely učenia, testovania a porovnávania účinnosti metód, vznikla v roku 1999 množina dát pomenovaná **1999 KDD Cup** [124], ktorá je v posledných rokoch jedna z najpoužívanejších množín pre testovanie účinnosti rôznych sieťových detekčných metód. *KDD Cup* obsahuje približne 4,9 miliónov vektorov spojení, ktoré sú označené za validné spojenie alebo útok. Z každého spojenia je extrahovaných 41 rysov (features). Táto sada bola ale kritizovaná [121], pretože dáta nie sú podobné tým, ktoré sú pozorované v bežnej sieťovej komunikácii. I keď niektorí vedci poukazujú na nové formy útokov, ako na rôzne variácie už známych foriem útokov [64], bolo v rámci *KDD Cup* (tzv. *Classifier Learning Contest*) poukázané, že to pravda nie je [48]. Postupom času veľa výskumných projektov zameraných na zvýšenie efektivity detekcie útokov v rámci *KDD Cup* nedosahovali dobré výsledky [62, 124, 53, 20] a vznikali projekty na rozšírenie a vynovenie tejto databázy, ako bol napríklad projekt *NoaH* [94].

DARPA IDS – Dátová sada **DARPA IDS** [129] bol vytvorený za účelom trénovania a testovania IDS technológií. Množina komunikácií ale bola generovaná pomocou SW, ktorý nie je verejne dostupný a nie je tak možné zistiť presnosť a autenticitu generovaného sieťového toku, tak isto chýbajú ďalšie informácie o uzloch siete, architektúre apod. Sada *DARPA* bola vytvorená v roku 1999 a v roku 2000 rozšírená o ďalšie simulované útoky. Tieto dátové sady sú používané od roku 1999 až do roku 2015 a slúžia pre porovnanie účinnosti rôznych detekčných metód a nástrojov založených na detekcii anomálií.

CDX 2009 – Dátová sada CDX 2009 [115] (*Cyber Defense Exercise*) pozostáva so zachyteného sieťového toku v rámci vytvorenej súťaže medzi útočiacimi tímami na pripravenú infraštruktúru. Cieľom bolo vytvoriť dátovú sadu útokov, ktorá je tvorená odchytenou TCP komunikáciou a popisom útokov (vzniknutých incidentov) zo záznamov IDS systému *Snort* [127]. Vytvorená infraštruktúra pozostávala z 3 fyzických systémov na ktorých bolo vybudovaných 8 rôznych virtuálnych infraštruktúr na ktorých boli umiestnené zraniteľné systémy (tri na každú podsieť). Počas štyroch dní útočilo približne 30 ľudí na tieto podsiete so zámerom infikovať, prípadne zničiť uvedené systémy. Na rozdiel od ostatných dátových sád, v cvičení CDX bol prítomný i tím, ktorý neustále v cieľovej sieti vytváral validný dátový tok.

V časti 2.2.4 sú uvedené IDS systémy, stručné zhrnutie použitých metód a porovnanie ich účinnosti voči uvedeným testovaným dátovým sadám. V nasledujúcom texte sú uvedené

aktuálne nástroje a použité metódy v detekcii anomálií zo sieťového toku. Tieto nástroje sú rozdelené na základe vstupných dát použitých metód. Prístupy môžeme rozdeliť na detekciu anomálií na základe atribútov paketov a detekciu dátovej časti analyzovaných paketov.

2.2.2 Detekcia anomálií na základe atribútov paketov

Detekčné metódy, ktorých vstupom sú atribúty paketov využívajú proces tzv. *features selection* – výber rysov vstupných dát, ktoré sú príznačné pre analyzovanú komunikáciu. Tieto rysy sú následne použité v metódach pre klasifikáciu komunikácie (či ide o útok alebo validnú komunikáciu).

V článku [49] autori uviedli detekčný systém **MINDS**, ktorý využíva techniky dolovania dát na automatizáciu detekčného procesu. Tento systém je postavený nad *NetFlow* [32] dátach (verzie 5), ktoré sú zbierané a analyzované v 10–minutových časových oknách. Hlavnými nevýhodami tohoto procesu je samotný *NetFlow*, ktorý neobsahuje dostatok informácií pre potreby ďalšej analýzy a na detegované udalosti je potrebný zásah človeka, ktorý dokáže overiť, či sa skutočne jedná o útok. *MINDS* pri dolovaní dát používa asociačné pravidlá s pravdepodobnosťou výskytu v pozorovaných anomálnych a validných spojeniach, tzn. výsledok klasifikácie je daný celkovou pravdepodobnosťou nájdených pravidiel (pomerom výskytu pravidiel v komunikácii označenej za útok a počte výskytov vo validnej komunikácii). I napriek spomenutým nevýhodám má údajne tento prístup dobré výsledky pri detekcii neznámych útokov (výsledky experimentov ale neboli zverejnené). V článku [118] bol uvedený nový pravdepodobnostný prístup, ktorý používa Markov reťazec pre modelovanie abnormálnych udalostí na sieťovom zariadení. Účinnosť navrhnutého systému bola otestovaná na uvedenej množine dát *DARPA 2000*.

V roku 2005 Moore et al. publikovali sadu dát [89] za účelom hodnotenia klasifikačných metód v detekcii sieťových anomálií. Bolo vytvorených a popísaných niekoľko množín dát, každá množina obsahuje niekoľko objektov, kde každý objekt je popísaný množinou rysov (tzv. diskriminátorov). Každý takýto objekt reprezentuje jeden tok medzi klientom a serverom. Jednotlivé diskriminátory sú použité ako vstup pre pravdepodobnostný klasifikačný algoritmus (nástroj nebol pomenovaný, pre účely tohoto textu bude používaný názov Moore). Pre klasifikáciu je použitá naivná forma bayesovského klasifikátora, ktorý vypočítava a posteriori pravdepodobnosť testovacej vzorky a vyberá najpravdepodobnejšiu triedu ako výsledok tejto klasifikácie. Pre trénovanie je celkovo použitých viac ako 200 rysov získaných zo sieťového toku, pre určenie distribučnej funkcie je použitá tzv. kernel funkcia [90] (pozri kapitolu 3.6.2). Celková presnosť metódy je približne 95 % z celkového počtu správne klasifikovaných tokov. Kernel je váhová funkcia (výstupom je váha) používaná na odhad funkcií hustoty náhodných atribútov, prípadne pri kernel regresii na podmienený odhad očakávanej hodnoty náhodných atribútov.

Na tento výskum nadviazal Auld a kol. [10], ktorý na základe Bayesovej metódy [90] navrhol klasifikačnú metódu založenú na bayesovskej neurónovej sieti. V porovnaní s naivnou bayesovou metódou dosiahol úspešnosť klasifikácie 99 % (o 4 % viac).

NIDES [7] je nástroj, ktorý monitoruje IP adresy a porty v čase bez trénovacej množiny a to za predpokladu, že v dlho trvajúcom časovom intervale nedochádza skoro k žiadnym útokom. Následná analýza porovnáva krátky časový úsek oproti hodnotám v rámci monitorovaného (dlhotrvajúceho) času. V prípade, že sa vzorka z krátkeho úseku (sekundy) výrazne líši, je táto komunikácia označená za útok.

Nástroj **ADAM** *(Audit Data and Mining)* [15] je zástupca, ktoré používajú pravdepodobnostný model klasifikácie na klasifikáciu útoku a validnej komunikácie. *ADAM* pou-

žíva metódu naivného bayesovského klasifikátora, ktorý klasifikuje analyzované dáta podľa pravdepodobnosti príslušnosti danej vzorky do triedy. Táto pravdepodobnosť je závislá na a priori pravdepodobnosti danej triedy a kombinácii pravdepodobností kolekcie asociačných pravidiel za predpokladu, že sú nezávislé (naivita u bayesovského klasifikátora). *ADAM* monitoruje IP adresy, porty, podsiete a TCP stav. Úspešnosť takejto klasifikácie je závislá na trénovacej sade (a priori pravdepodobnosť).

Systém **SPADE** *(Statistical Packet Acceptance Defense Engine)* [130] používa štatistickú analýzu založenú na modelovaní sieťového toku pomocou hierarchických štruktúr, ktoré sú reprezentované stromovou štruktúrou, ktorá z listov agreguje štatistické informácie do koreňov. Týmto prístupom sa dá jednoducho a rýchlo detegovať DoS a DDoS útoky na sieťovej vrstve, systém ale neslúži na detekciu iných útokov a obsahuje iba informácie o IP adrese, protokole, porte a údaje o veľkosti, ktoré sú použité pri analýze. Tá je závislá na nastavenom parametri, ktorý udáva kapacitu siete. Ďalším z predstaviteľov detektorov DoS a DDoS útokov pomocou stromovej štruktúry je **MULTOPS** *(MUltilevel Tree for Online Packet Statistics)* [54]. Tento nástroj na rozdiel od *SPADE* je priamo určený pre implementáciu do sieťových smerovačov. *MULTOPS* funguje na princípe monitorovania štatistických informácií o prebiehajúcich komunikáciách medzi dvoma bodmi v sieti. V prípade, že počet prichádzajúcich spojení je neadekvátny počtu odchádzajúcich spojení, sú tieto spojenia označené ako útok a zahodené na výstupe zo sieťového smerovača. *MULTOPS* používa stromovú štruktúru, kde koreň stromu obsahuje tabuľku vlastných potomkov na sub-domény, tie sa ďalej delia na menšie sub-domény do hĺbky štvrtej úrovne.

Ďalším zástupcom výskumných nástrojov pre detekciu útokov v sieťovom toku na základe štatistických metód, je systém pozostávajúci z dvoch častí: **PHAD** *(packet header anomaly detection)* a **ALAD** *(application layer anomaly detection)* [82]. Na rozdiel od nástrojov *ADAM*, *NIDES* alebo *SPADE*, nemonitoruje iba IP adresy a porty zdrojovej a cieľovej stanice, ale do analýzy u *PHAD* pridáva celú hlavičku TCP, UDP alebo ICMP protokolu a do *ALAD* pridáva kľúčové slová aplikačného protokolu (napr. pre HTTP: GET, Host: atď.), zdrojovú a cieľovú IP adresu, porty a TCP atribúty. V rámci učiacej fáze sa u aplikačnej i paketovej analýzy stanovuje (učí) rozsah jednotlivých atribútov, ktoré sú zhlukované do tried. Následne sa pomocou pravdepodobnostnej analýzy stanovuje príslušnosť do tried podobne ako ostatné spomínané nástroje. Celkovo ale nástroj podľa [82] na dátovej množine *DARPA IDS* dosiahol iba 39 % senzitivitu s 41 % precíznosťou.

Ďalšie výskumné projekty a nástroje na detekciu útokov pomocou analýzy anomálií v sieťovom toku je možné uviesť napríklad **MADAM ID** *(Mining Audit Data for Automated Models for Intrusion Detection)* [78], framework, ktorý získané sieťové dáta spracuje dolovacími algoritmami pre extrakciu rysov a asociačných pravidiel pre klasifikačné algoritmy, ďalšími zástupcami IDS systémov, ktoré analyzujú atribúty komunikácií sú napr. **EMERALD** [103] a **NetSTAT** [132]. Viac informácií o detekčných systémoch je možné čerpať v uvedenej literatúre [83, 93, 94].

2.2.3 Detekcia anomálií na základe obsahu

V prípade zahrnutia dátovej časti paketu do analýzy sieťového toku vznikajú nástroje, ktoré majú veľmi vysokú úspešnosť detekcie nešifrovaných spojení (napr. HTTP, TELNET, DNS apod.). Jedným z reprezentantov detekcie anomálií v dátovej časti paketov je **PAYL** [136]. Tento nástroj analyzuje štatistickú distribúciu bytov v paketoch. V trénovacej fáze vytvára profily komunikácie, ktoré sú reprezentované distribúciou bytov v dátovej časti paketoch. Pre dátovú časť každého prichádzajúceho paketu je vypočítaná priemerná frekvencia bytov

a štandardná odchýlka. Pri vyhodnotení je počítaná distribúcia bytov aktuálneho spojenia a jeho vzdialenosť (použitá *Mahalanobisova vzdialenosť* [43]) od štatistickej distribúcie daného profilu. Pravdepodobnosť, že ide o útok je vypočítaná na základe tejto vzdialenosti (čím väčšia vzdialenosť, tým pravdepodobnejší je útok). V rámci experimentov nad dátovou sadou *DARPA IDS* dosiahol nástroj *PAYL* 58,8 % senzitivitu pri hodnote false-positive nižšej ako 1 %. Ale pri analýze nešifrovanej HTTP komunikácie na porte 80 odhalil 100 % útokov.

Na báze nástroja *PAYL* boli neskôr postavené ďalšie nástroje, ktoré *PAYL* modifikovali. Príkladmi týchto nástrojov sú **POSEIDON** *(Payl Over Som for Intrusion DetectiON)* [21] a **ANAGRAM** [134]. *POSEIDON* je nadstavbou nad *PAYL* systémom, kde v prvom module je implementovaný klasifikátor založený na samo-organizovaných mapách (z angl. *Self-Organizing Maps*, alebo skrátene *SOM*), v druhom module je potom použitá modifikovaná verzia *PAYL* nástroja. *POSEIDON* zvýšil senzitivitu *PAYL* nástroja z 58,8 % na 73,19 % pri zachovaní hranice 1 % false-positive. *ANAGRAM*, nástroj od tvorcov *PAYL*, ktorý na rozdiel od ich predchádzajúceho nástroja nie je postavený na výpočtoch frekvenčných distribúcií bytov v obsahu validných paketov (a ich následnej detekcie pomocou výpočtu vzdialenosti od validných profilov), ale zameriava sa na útoky z ktorých extrahuje a ukladá n-gramy do *Bloomových filtrov*. Tým sa zaručí čiastočná príslušnosť (pravdepodobnosť, že do danej množiny nepatrí) analyzovaného paketu do množiny extrahovaných častí paketov zo zachytených útokov. Výsledky experimentov je ťažké posúdiť, pretože boli vykonané na vlastných dátach nešifrovanej webovej komunikácie a samotné testovanie prebiehalo len voči naučeným útokom.

Jedným z posledných (najnovších) nástupcov kategórie detekčných systémov založených na anomáliách je **Octopus-IIDS** *(Octopus Intelligent Intrusion Detection System)* [81], ktorý je postavený nad algoritmami neurónových sietí, konkrétne Kohonenových sietí [96] a **SVM** *(Support Vector Machines)* [58, 37]. Systém je zložený z dvoch vrstiev. Prvá, pomenovaná ako klasifikátor, je zodpovedná za zbieranie dát a ich klasifikovanie do štyroch kategórií: DoS (útok na dostupnosť sieťových služieb), U2R (útok na získanie administrátorského prístupu root z lokálneho účtu), Probe (skenovanie) a R2L (útok na získanie lokálneho účtu zo siete). Druhá vrstva IDS systému je rozdelená do štyroch SVM sietí podľa kategórií, každá zodpovedná za určenie, či daná komunikácia patrí do danej kategórie. Vstupom každého SVM v danej kategórii je množina rysov, ktoré sú dopredu určené (napr. IP adresa, trvanie komunikácie, typ služby apod.). Experimenty s *Octopus-IIDS* boli vykonané nad dátovou sadou *KDD Cup* s rôznymi formami algoritmov s celkovou priemernou úspešnosťou 97,40 %.

Ďalšie metódy klasifikácie je možné vidieť v článku od autorov Haque a Talal [61], ktorí prezentovali hybridný model IDS (pre účely tejto práce pomenovaný podľa názvu článku **AHM-NID**) s porovnaním úspešnosti detekcie s aktuálnymi IDS systémami nad *KDD Cup* sadou dát. Podobne ako *Octopus-IIDS* rozdelili jednotlivé útoky do kategórií (podľa sady dát) a porovnali úspešnosť detekcie naivného bayesovského klasifikátora, metód **PART** [29], **Random Forest** [25], **Grading Classifier** [97], **Adaboost** [51] a **IBK** [2]. Tak isto v práci porovnávajú rôzne metódy redukcie rysov analyzovaných dát a ďalších metód pre úpravu vstupných dát do klasifikačných algoritmov. Podľa experimentov je úspešnosť klasifikácie dát (nad *KDD Cup*) 95,18 % pre naivný bayesovský klasifikátor, 98,75 % pre Random Forest metódu a 99,60 % pre PART algoritmus.

Autori v článku [8] predstavili nástroj **PCkAD** *(Packet Chunk Anomaly Detector)*, ktorý je určený na detekciu anomálií (útokov) v nešifrovanej komunikácii pomocou metódy

n-grams[8] (analýza distribúcie bytov v dátovej časti paketov), ktorý je posilnený znalosťou protokolu. Táto znalosť je získavaná analýzou tzv. *chunkov*, reťazcov konštantnej dĺžky (30 bytov), ktoré obsahujú reťazce, ktoré sú pre daný protokol charakteristické (napr. *GET* a *POST* pre HTTP). Protokol je reprezentovaný číslom portu a/alebo počtom chunkov protokolu, ktoré obsahuje komunikácia. Výsledky experimentov nad dátovou množinou *1999 DARPA IDS* dosiahol 100 %, ale pokusy boli vykonané iba nad FTP protokolom a preto sú neporovnateľné s výsledkami ostatných IDS a ADS systémov.

Nástroj **McPAD** [100] nadviazal na uvedený *PCkAD*. *McPAD* používa pozmenenú metódu n-grams, konkrétne *2-grams*, ktorá ale neporovnáva frekvenciu výskytov dvojíc bytov, ale využíva tzv. *sliding window* (okno o veľkosti v, ktoré sa posúva od začiatku po koniec reťazca v krokoch a pri každom posunutí analyzuje viditeľné byty dané veľkosťou okna, tzn. v *bytov*). Samotná klasifikácia je potom vykonávaná algoritmom **SVN** [38] s bayesovským klasifikátorom. Výsledky nástroja je ťažké interpretovať, pretože popri dátovej sade *DARPA* použili vlastnú dátovú sadu nad ktorou robili experimenty a sadu *DARPA* rozdelili na podmnožiny podľa typu útokov. Článok uvádza dosiahnutú pravdepodobnosť detekcie 95 %.

Ďalší zástupca detekčných nástrojov zameriavajúcich sa na nešifrovanú komunikáciu je predstavený v článku od Ariu Davide a kol. [9] v ktorom popisujú nástroj **HMMPayl** *(Intrusion Detection System based on Hidden Markov Model)*, ktorý sa sústreďuje na detekciu anomálií v nešifrovanom HTTP aplikačnom protokole. Autori mali výhrady voči aktuálnym metódam detekcií útokov na webové aplikácie (založené na analýze štatistickej distribúcie bytov v dátovej časti paketov), ktoré sú neefektívne voči útokom, ktoré nemenia štatistické rozloženie bytov alebo ho menia iba minimálne (napr. XSS alebo SQL Injection). *HMMPayl* vykonáva analýzu paketov v troch krokoch:

1. extrakcia rysov z komunikácie,
2. analýza vzorov a
3. samotná klasifikácia pre určenie útoku na monitorovanú webovú aplikáciu.

Vzhľadom na to, že *HMMPayl* modeluje dátovú časť paketov HTTP komunikácie, táto komunikácia môže nadobúdať rôznu dĺžku (rôzna dĺžka tzv. payloadu), to spôsobuje komplikácie pri klasifikácii a preto bol vytvorený algoritmus náhodne sa posúvajúceho okna (z angl. *sliding window*), ktorý vytvára sekvencie bytov, ktoré sú postupne analyzované. Experimenty boli testované nad *DARPA IDS* dátovou množinou. U útokov, ktoré posielajú na cieľový systém shell-code, ktorý obsahuje binárne dáta, bola detekcia 100 %. U útokov, ktoré obsahujú iba textové dáta (v textovom HTTP protokole) sa líšili podľa útoku medzi 84-92 %.

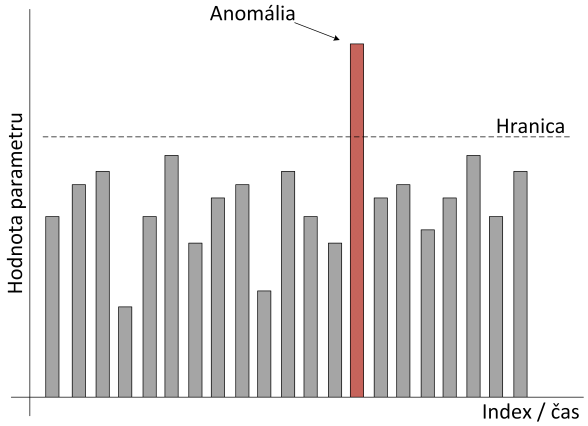
Jedným zo zástupcov hybridných IDS systémov je **RT-MOVICAB-IDS** *(MObile VIsualisation Connectionist Agent-Based IDS)* [63]. Tento nástroj kombinuje metódy neurónovej siete (z angl. *Artifical Neuron Network – ANN*), tzv. *Case-based Reasoning* (*CBR*) a *multi-agentného systému* (*MAS*). Hlavným cieľom autorov tohoto nástroja je vizualizácia relevantných dát administrátorovi pre manuálnu analýzu potenciálnych incidentov. Nástroj rozhodovací proces pri klasifikácii presúva z algoritmickej roviny na ľudskú a to tak, že sa snaží optimalizovať vizualizáciu nazbieraných, sčasti analyzovaných, dát a klasifikáciu ponecháva na človeku – analytikovi.

[8] Metóda n-grams v rámci tejto práce je použitá v zmysle výberu n po sebe idúcich slov alebo iných jednotiek (písmen, bytov apod.)

2.2.4 Prehľad metód na detekciu útokov na sieti

V článku [93] je uvedený prehľad klasifikačných metód za roky 2004 – 2007. V tomto článku sú vytvorené štyri kategórie klasifikačných algoritmov a to prístup **zhlukovania**, **učenie s učiteľom**, **hybridné** a **porovnávacie algoritmy**. Z aktuálneho prehľadu metód, ktoré sa sústreďujú na detekciu anomálií v sieťovom toku je možné tieto metódy rozšíriť o skupiny, ktoré lepšie popisujú charakteristiky použitých metód a techník a rozdeliť uvedené metódy do nasledujúcich skupín.

- **Štatistické** – založené na štatistických metódach, zvyčajne modelovaním validného správania a detekciou na základe odklonu od tohto modelu (napr. *NIDES*, *PAYL*), prípadne sledovaním konkrétneho parametru (napr. veľkosť paketu) a stanoveniu hranice (z angl. *threshold*) ako je vidieť na obrázku 2.2. Po prekročení stanovenej hranice je skúmaný prvok označený za anomáliu.



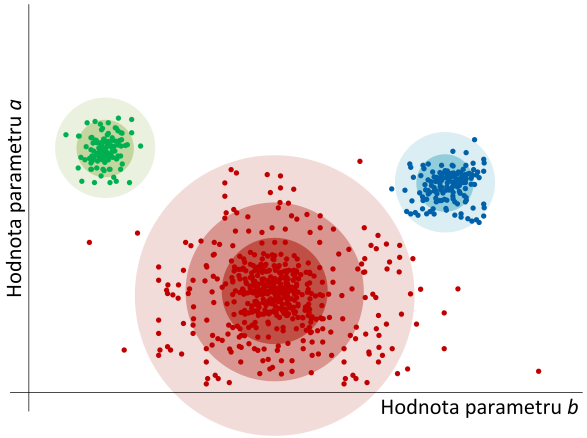
Obr. 2.2: Threshold / hranica detekcie anomálie nad analyzovaným parametrom.

- **Klasifikačné** – pri klasifikácii je vstup rozdelený do dvoch alebo viacerých tried (validná komunikácia a útok) a cieľom klasifikačných metód je vytvorenie modelu, ktorý ďalšie neklasifikované vstupy správne klasifikuje do tried. Jedným z predstaviteľov je *Moore*, ktorý pomocou vytvorených metrík sieťového toku klasifikoval analyzovanú komunikáciu bayesovským klasifikátorom (pravdepodobnostným ohodnotením, pre viac informácií pozri kapitolu 3.6.2).

- **Učenie s učiteľom** – metódy učenia pri ktorých je danej technike/metóde dodaná sada vstupov (príklady validných komunikácií a útokov) a očakávaný výstup. Cieľom týchto algoritmov je nájdenie pravidla, ktoré mapuje vstupy na očakávané výstupy (príkladom môže byť *McPAD* alebo *Octopus-IIDS*, ktorý používa SVM metódu pre klasifikáciu tokov na validné a útoky štatistickou analýzou obsahu paketov).

- **Zhlukovacie** – zhlukovanie (z angl. *clustering*) je v kontexte strojového učenia metóda hľadania štruktúry dát bez jej predchádzajúcej znalosti. Klasifikačné metódy určené na detekciu útokov používajú zhlukovanie na vyhľadanie čŕt v parametroch sieťovej komunikácie v rámci útokov. Označené dáta útokov tak môžu byť analyzované pomocou algoritmov zhlukovania na nájdenie parametrov najlepšie určujúcich či ide o útok alebo o validnú komunikáciu. Táto znalosť sa následne použije pri samotnej

klasifikácii (i keď zhlukovanie nie je klasifikačný algoritmus). Na obrázku 2.3 je zobrazený náčrt princípu zhlukovania na základe distribúcie, podľa ktorej je možné stanoviť vzájomnú koreláciu/závislosť týchto parametrov. Jednotlivé zhluky môžu zodpovedať napr. protokolom, rôznym sieťovým aplikáciám, typom systémov, ale i napr. typu útoku (prvok nezodpovedajúci žiadnemu zo zhlukov môže byť označený za anomáliu).



Obr. 2.3: Náčrt zhlukovania podľa 2 parametrov na základe distribúcie.

- **Porovnávacie** – alebo tiež vyhľadávacie, v odchytenom sieťovom toku vyhľadávajú reťazce známych útokov, prípadne ich indikátorov. Príkladom je systém Buttercup, ktorý v dátovej časti paketov vyhľadáva rozsahy návratových hodnôt, ktoré sú indikátorom payloadu u buffer overflow útokoch. Ďalšie porovnávacie techniky a metódy (napr. *CTCP*, *Poseidon*, *Anagram* atď.) vyhľadávajú časti už analyzovaných útokov v zachytenej komunikácii.
- **Hybridné** – hybridné algoritmy používajú kombináciu detekčných techník a metód na rôznych úrovniach. Príklad môže byť kombinácia detekčných metód na základe atribútov paketu a na základe ich obsahu (napr. *PHAD/ALAD*).

2.3 Zhodnotenie aktuálneho stavu v detekčných metódach

V tabuľke 2.2 je zobrazený prehľad detekčných nástrojov a techník podľa roku ich zverejnenia. V rámci výskumných prác je možné nájsť i odlišné hodnoty (hlavne v publikáciách od iných autorov, ktorí testovali niektoré z uvedených techník/nástrojov, v rámci tejto práce ale nie je možné uviesť úplný zoznam uverejnených nástrojov a experimentov). Pri každom nástroji je uvedená hodnota dosiahnutých výsledkov (v prípade, že bola zverejnená) a to úspešnosť detekcie. Tieto hodnoty sú prevzaté zo zdrojov (väčšinou vedeckých článkov) uvedených pri popise jednotlivých nástrojov a nebola overená ich pravdivosť.

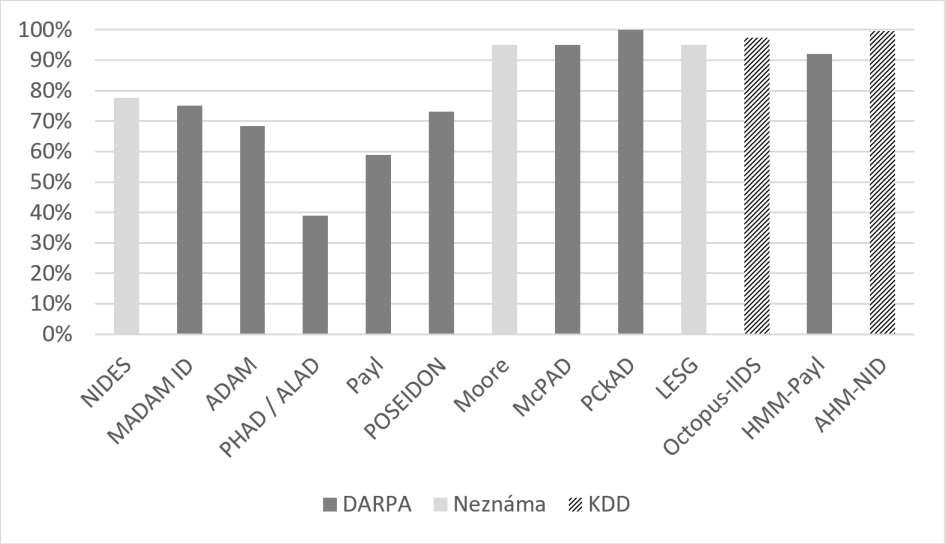
Nástroj	**Rok**	**Použitá metóda**	**Výsledok**	**DB**
NIDES	1995	NIDS, Štatistické modely	77,69 %	-
NetSTAT	1998	NIDS, jednoduché pravidlá v rámci atribútov paketov	-	-

Nástroj	Rok	Použitá metóda	Výsledok	DB
MADAM ID	2000	Framework pre štatistickú klasifikáciu	75 %	DARPA
MULTOPS	2001	Štatistická analýza na základe počtov paketov	-	-
ADAM	2001	Pravdepodobnostný model	68,30 %	DARPA
PHAD / ALAD	2002	Hybridný prístup so štatistickou analýzou paketov a DPI aplikačných protokolov	39 %	DARPA
BRO	2003	NIDS, jednoduché pravidlá v rámci paketov	-	-
Payl	2004	Štatistická distribúcia bytov v obsahu paketov	58,80 %	DARPA
CTCP	2004	Vyhľadávanie konkrétnych vzorov pri buffer overflow detekcii v prerušenej komunikácii (ako MitM)	-	-
Buttercup	2004	Filtrácia obsahu paketov na základe hľadaného rozsahu návratových adries pri buffer overflow.	-	-
SHIELD	2004	Manuálne modelovanie známych zraniteľností stavovým grafom	-	-
MINDS	2004	Detekcia sieťových útokov na základe asociačných pravidiel	-	-
POSEIDON	2005	Vyhľadávanie n-gramov v obsahu komunikácie a klasifikácia na základe existencii komunikácie v Bloom filtri.	73,19 %	DARPA
ANAGRAM	2006	Detekcia útokov v nešifrovanej webovej komunikácii pomocou analýzy n-grams dát paketov použitím Bloomových filtrov	-	-
Moore	2007	Pravdepodobnostná analýza nad štatistickými informáciami z atribútov paketov	95 %	-
McPAD	2009	Nešifrovaná komunikácia pomocou n-gramov a SVN	95 %	DARPA
PCkAD	2009	Nešifrovaná komunikácia pomocou n-gramov so štatistickou analýzou výskytu n-gramov v zachytených útokoch	100 %	DARPA
LESG	2010	Generátor signatúr zo zraniteľností buffer overflow pomocou analýzy dĺžky polí aplikačných protokolov	95 %	-
SPADE	2010	Detekcia DoS útokov štatistickými metódami	-	-
Octopus-IIDS	2010	Štatistická analýza obsahu paketov pomocou SVM	97,40 %	KDD
HMM-Payl	2011	Štatistická analýza pomocou skrytého Markovho modelu	84 - 92 %	DARPA

Nástroj	Rok	Použitá metóda	Výsledok	DB
RT-MOBICAB-IDS	2013	Vizualizácia relevantných dát použitím MAS, ANN a CBR	-	-
AHM-NID	2015	Klasifikácia sieťového toku pomocou PART algoritmu	99,60 %	KDD

Tabuľka 2.2: Porovnanie nástrojov na detekciu útokov zo sieťového toku.

Vzhľadom na to, že tieto nástroje nie sú verejné, nie je ani možnosť porovnania ich účinnosti nad jednou vstupnou dátovou množinou a porovnanie tak nie je objektívne (každý nástroj používa vlastné vstupné dáta alebo verejné dátové množiny *DARPA* alebo *KDD Cup*).



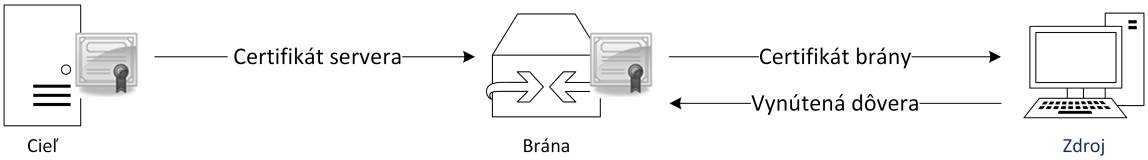
Obr. 2.4: Porovnanie dostupných uvádzaných výsledkov detekčných metód podľa účinnosti a použitej vstupnej dátovej sady.

Problém v porovnaní detekčných metód. Problém je spôsobený niekoľkými faktormi. Približne polovica nástrojov neuvádza výsledky (alebo výsledky nie sú verejne prístupné, či jednoznačné) v dostupnej literatúre a ich porovnanie s ostatnými metódami je tak nemožné. Ďalej sú výsledky kvalitatívne neporovnateľné keď jednotlivé nástroje boli testované na rôznych databázach. Niektoré z metód naopak pri použití databázy testujú nástroj len na jej časti (napr. nástroj *PCkAD* používa len nešifrovanú FTP komunikáciu) alebo si na testovanie autori vytvorili (napr. *Moore*) vlastnú dátovú sadu.

V rámci tejto kapitoly bol predstavený prierez dvadsiatich rokov výskumu a vývoja metód a techník na detekciu sieťových útokov, ktoré prešli dlhým časovým obdobím, ktoré prinieslo radu zmien ako v útokoch na sieťové systémy, tak v použitých metódach a ako je vidieť na obrázku 2.4, i účinnosť týchto nástrojov sa s časom zväčša zvyšovala. Je nutné ale podotknúť, že v oblasti použitých metód a techník existuje niekoľko problémov, ktoré vplývajú na ich účinnosť.

2.3.1 Šifrovanie sieťovej komunikácie

Veľa sieťových detekčných nástrojov je odkázaná na nešifrovanú komunikáciu (zväčša IDS systémy založené na signatúrach). Použité metódy sa zameriavajú na vyhľadávanie podreťazcov v dátovej časti paketov analyzovaného spojenia a ich porovnávanie s množinou známych signatúr. Vzhľadom na to, že tendencia šifrovania komunikácie a zavádzanie šifrovania i v interných sieťach organizácií pridáva tomuto prístupu nutnosť SSL terminácie[9] (pozri obrázok 2.5) na prvku v ceste medzi koncovým užívateľom a systémom na Internete. Tento prístup má ale svoje vlastné problémy, ktorých popis a riešenie sú nad rámec tejto práce.



Obr. 2.5: SSL terminácia s explicitnou dôverou v zastupujúci certifikát brány pre certifikáty všetkých SSL spojení na externé servery.

Metódy, ktoré nie sú založené na DPI (hĺbková inšpekcia paketov, z angl. *Deep Packet Inspection*), v rámci ktorej je potrebné vidieť nešifrovanú komunikáciu, sú voči tomuto problému odolné v prípade, že vstupom metód sú len hlavičky paketov. Šifrovanie má síce určitý vplyv na veľkosť a rozloženie dát v rámci komunikácie (napr. zarovnanie použitého šifrovacieho algoritmu alebo nárast veľkosti prenesenej komunikácie), ale tento vplyv je zanedbateľný.

2.3.2 Neexistencia obecnej dátovej sady s útokmi

V rámci prehľadu dátových sád (pozri kapitola 2.2.1), ktoré boli využívané uvedenými detekčnými nástrojmi pre účely výskumu a testovania boli popísané sady *DARPA 1999* a *DARPA 2000* [129], *1999 KDD Cup* [124] a *CDX 2009* [115]. V článku [125] autori poukázali na problémy s evaluáciou sieťových metód na detekciu útokov pomocou IDS systémov. V článku bolo použitých 276 článkov medzi rokmi 2000 a 2008. Štúdia ukázala, že v 70 % článkov boli použité na experimenty verejne dostupné databázy, 32 % vlastné dátové množiny, 9 % simulované prostredie a v 7 % prác boli nasadené detekčné systémy do reálneho prostredia. Z verejných dátových sád v 24 % analyzovaných článkov bola použitá *DARPA* a 28 % *KDD* dátová sada. Iba v 6 % bola použitá iná dátová množina a v 15 % bola pôvodná sada zmenená pridaním vlastných záznamov. Zvyšok výskumov použilo sady, ktoré sú určené na detekciu útokov na systémoch (tzv. *host-based detection*). Článok poukazuje na rozšírenú kritiku voči týmto dátovým sadám z ktorých je potrebné zohľadniť nasledujúce faktory:

- Dátové sady sú vytvárané v simulovaných prostrediach a neodzrkadľujú podmienky v reálnom prostredí.

[9]SSL terminácia je proces ukončenia SSL šifrovaného spojenia na systéme. V rámci terminácie spojení je zdroj nútený použiť na všetku komunikáciu certifikáty terminujúceho prvku, prípadne musí zdieľať súkromné kľúče s bránou. Komunikácia medzi terminujúcim prvkom, ako aj medzi prvkom a cieľovým systémom je šifrovaná, ale týmto spôsobom je možné v rámci terminujúceho prvku vidieť obsah komunikácie.

- Experimenty nad dátovými sadami nie sú dostatočne popísané – chýbajú základné údaje, ako špecifikácia dátových množín použitých pri učení a pri testovaní, výber konkrétnych rysov a parametrov apod.

- Pomer validných záznamov a útokov je nevyvážený (napr. *KDD* dátová sada obsahuje 81 % záznamov útokov).

Tento problém je ťažko riešiteľný v prípade neexistencie verejnej dátovej sady útokov a validnej komunikácie, ktorá bude dostatočne popísaná, bude obsahovať toky z reálneho prostredia (prípadne viac prostredí pre čo najväčšiu diverzitu záznamov), bude verejne dostupná a bude definovať postup použitia dátovej sady tak, aby výsledky jednotlivých systémov mohli byť porovnateľné (viac k systému hodnotenia v kapitole 2.3.4). Dátové sady, ktoré vznikajú pri rôznych hackerských súťažiach, v rámci ktorých hráči (tímy hackerov) útočia na systémy protihráčov, pozostávajú z dát, ktoré sú typicky generované v krátkom čase veľkým množstvom zdrojov, ktoré vo veľkej miere útočia na cieľové systémy. V týchto sadách (*DARPA*, *KDD Cup*) chýbajú bežné reálne sieťové spojenia, živé spojenia s Internetom apod. V reálnom prostredí sú očakávané skôr automatické pravidelné útoky malware, než sieť presýtená útokmi bez validnej komunikácie. Túto situáciu nezlepšuje nezverejnenie použitého algoritmu a parametrov použitých pri klasifikácii tak, aby sa dal v prípade potreby pokus opakovať mimo pôvodnú výskumnú prácu.

2.3.3 Malá komplexnosť detekčných nástrojov

Ďalším problémom v analyzovaných detekčných metódach je ich veľmi úzke zameranie na konkrétny problém alebo oblasť a v porovnaní s ostatnými detekčnými metódami a nástrojmi sa tieto práce stávajú neporovnateľnými. Niektoré nástroje sa zameriavajú na inšpekciu dátovej časti paketov voči signatúram zachytených útokov. Tieto metódy sú účinné voči špecifickým útokom a je ťažké ich výsledky porovnávať s metódami, ktoré sú výrazne komplexnejšie (s rastúcou komplexnosťou – pokrytie širokého spektra typov útokov klesá účinnosť detekcie alebo rastie počet nesprávne detegovaných útokov) a to hlavne v prípade, že pri testovaní týchto úzko zameraných nástrojov je použitá iba určitá časť testovacej sady, ktorá vyhovuje danému experimentu.

2.3.4 Nejednotný systém hodnotenia účinnosti detekcie

V kapitole 1.3 sú uvedené pojmy *senzitivita*, *špecifickosť*, *precíznosť*, *účinnosť* a *presnosť* klasifikácie. Tieto štatistické funkcie sa používajú na určenie účinnosti metód (nemyslí sa účinnosť ako uvedená funkcia, ale obecný pojem reprezentujúci kvalitu detekčných metód), ale často môžu byť náchylné na skreslenie štatistických dát, hlavne pri použití nerovnomerných vzoriek (napr. nepomer vzoriek útokov k validnej komunikácii). Pre lepšie výsledky vo vedeckých prácach a článkoch je tak možné použiť zvolenú štatistickú funkciu, ktorá nereflektuje najhorší parameter (z *TP, FP, TN, FN*). Napríklad pri vysokom počte nesprávne klasifikovaných útokov (FP) je možné použiť *senzitivitu* klasifikácie, ktorá zobrazuje pomer správne klasifikovaných validných komunikácií ku všetkým validným komunikáciám.

Tento problém môže byť vyriešený vytvorením dátovej sady pre experimenty so sieťovými detekčnými metódami, ktorej použitie pre výskumné účely bude podmienené (prípadne odporučené) dodaním výsledkov z experimentov v danom formáte. Ako vhodný formát sa zdá byť tzv. *F-score* [119, 55].

2.4 Zhrnutie

Metódy a techniky na detekciu útokov zo sieťového toku je možné rozdeliť na *detekciu založenú na signatúrach* a *detekciu založenú na anomáliách*. Signatúry môžu byť tvorené *bitovým reťazcom* alebo *pravidlami*. Tieto metódy ale majú veľkú nevýhodu v nutnosti vytvárania a udržovania databáz signatúr voči ktorým sa porovnáva sieťový tok. Presne špecifikovaná signatúra má výhodu v malom množstve chybných hlásení o útoku – false-positive, ale ich manuálne vytváranie je problematické a môže trvať príliš dlho. Z toho dôvodu vznikali metódy na automatické generovanie signatúr, ktoré je možné rozdeliť na *signatúry založené na exploite* a *signatúry založené na zraniteľnosti* a líšia sa procesom ich generovania. Automatické metódy na generovanie signatúr ale majú problémy pri určovaní, či ide naozaj o útok. Tento problém adresujú systémy s názvom honeypot, ktoré dokážu odhaliť útok na systém a v prípade útokov na pretečenie zásobníka – buffer overflow, vykazujú vysokú úspešnosť detekcie. Okrem výskumných prác okolo detekcie týchto útokov na honeypot systémoch, vznikli i práce zameriavajúce sa na detekciu týchto útokov v rámci sieťového toku, ale väčšinou ide o detekciu na základe obsahu paketov, ktorá je v prípade šifrovaných spojení neúčinná a v prípade, že sa podoba útoku zmení (bitová reprezentácia) dajú sa tieto metódy úplne obísť. Tento problém sa snažia adresovať techniky a metódy, ktoré sa sústreďujú na detekciu anomálií a tie je možné rozdeliť na detekciu na *základe atribútov paketov* a na *základe obsahu paketov*. Ich úspešnosť v detekcii sieťových útokov je obecne vyššia a v prípade atribútov paketov často imúnna voči šifrovaniu a rôznym obfuskačným technikám. Princíp detekcií anomálií na základe atribútov paketu je v metódach strojového učenia, ktoré je možné rozdeliť na základe uvedených vedeckých prác na *štatistické, zhlukovacie, klasifikačné, učenie s učiteľom, porovnávacie* a *hybridné*. Úspešnosť týchto metód je ale problematické skúmať a analyzovať z niekoľkých uvedených dôvodov medzi ktorými je najzásadnejšia neexistencia obecnej dátovej sady s útokmi (rôzne práce používajú rôzne dátové sady pre experimenty), nad ktorou by boli otestované všetky uvedené metódy a nástroje a zároveň uvedené výsledky klasifikácie by boli uverejnené jednotným systémom hodnotenia.

V nasledujúcich kapitolách je uvedená vlastná časť práce, ktorá sa zameriava na vyriešenie (niektorých) uvedených problémov a to návrhom architektúry sieťového detekčného systému. Návrh sa opiera o fundamentálnu časť systému a to definíciu obecných sieťových metrík na popis správania analyzovaných spojení.

Kapitola 3

Metriky paketových sietí

Definícia sieťových metrík sa bude v rámci tejto práce zameriavať na paketové siete, ktoré sú v dnešnom svete najrozšírenejšie. Paketové siete sú siete, kde každá informácia posielaná medzi zdrojom a cieľom je rozdelená na bloky, nazývané pakety, ktoré sú posielané po médiu, ktoré je zdieľané medzi ďalšími spojeniami. Každý paket obsahuje hlavičku s informáciami pre sieťové prvky a dátovú časť, kde nesie zasielanú informáciu. Paketové siete je možné rozdeliť na dve hlavné podskupiny:

- **Nespojované** (z angl. *connectionless*) – každý paket obsahuje všetky informácie potrebné pre ustanovenie spojenia a je smerovaný individuálne, čo môže spôsobovať rozdielnu cestu pre viacero paketov alebo príchod v inom poradí ako boli odosielané. Príkladom nespojovaných protokolov je *Ethernet*, *IP* alebo *UDP* protokol.
- **Spojované** (z angl. *connection-oriented*) – pred samotným zasielaním paketov je potrebné ustanoviť spojenie (dohodnúť sa na vhodných parametroch spojenia), pakety obsahujú identifikáciu spojenia (namiesto adresácie). Dôraz je kladený na doručenie paketov v stanovenom poradí s detekciou chýb. Príkladom spojovaných protokolov sú *X.25*, *Frame Relay*, *MPLS* alebo *TCP*.

Ďalšími príkladmi paketových sietí sú *GPRS*, *I-mode*, *IPX/SPX* atď. Pre zachovanie všeobecnosti je definovaný minimalistický sieťový paketový protokol, ktorý spĺňa len základné požiadavky pre prenos informácií a nad týmto minimalistickým protokolom sú definované sieťové metriky a ďalšie základné mechanizmy, ktoré budú potrebné pri analýze spojení a detekcii útokov.

Kapitola definuje minimalistický sieťový protokol nad ktorým je vytvorený formálny systém charakterizujúci sieťové spojenia a sústava metrík, ktorých výstupom je signatúra analyzovaného spojenia. Časť kapitoly je venovaná rozšíreniu minimalistického protokolu na TCP/IPv4 architektúru a prispôsobenie navrhnutého systému jej protokolom a parametrom sieťového toku. V závere kapitoly je uvedená definícia kontextu spojenia a sú načrtnuté metódy a algoritmy analýzy navrhnutých signatúr so zameraním na klasifikáciu tokov na útoky a validnú komunikáciu.

3.1 Minimalistický sieťový paketový protokol

Pre účely tejto práce sa sústredíme na podmnožinu protokolov paketových sietí a budeme uvažovať iba o spojovaných stavových sieťových protokoloch so spoľahlivým prenosom dát.

V rámci tejto úvahy je potrebné definovať atribúty paketov, ktoré sú potrebné pre definíciu takéhoto protokolu.

- **Poradové číslo (id)** – celé číslo, určuje pozíciu paketu v postupnosti, pre účely zoradenia paketov a rekonštrukcie komunikácie.
- **Veľkosť paketu (len)** – počet bitov paketu vrátane hlavičky reprezentovaný celým nenulovým číslom.
- **Zdrojová adresa paketu (src)** – adresa uzlu siete z ktorého bol paket odoslaný, zdrojová adresa je bitový reťazec konečnej dĺžky.
- **Cieľová adresa paketu (dst)** – bitový reťazec konečnej dĺžky jednoznačne identifikujúci cieľový uzol siete, pre ktorý je paket určený.
- **Dátová časť paketu (data)** – tzv. payload, dáta, ktoré sú určené pre vyššiu (napr. aplikačnú) vrstvu reprezentované postupnosťou bitov konečnej dĺžky.

Uzlom siete sa rozumie zariadenie, ktoré dokáže na sieti komunikovať, prijímať a odosielať pakety. Ďalej budeme predpokladať, že daná zdrojová i cieľová adresa uzlu je v rámci siete unikátna (neexistuje viac uzlov s rovnakou adresou). Ďalšie atribúty sa môžu líšiť na základe použitého protokolu, protokoly jednotlivých vrstiev môžu pridávať povinné i nepovinné atribúty podľa ich účelu. Pre účely tejto práce budú zatiaľ uvažované iba uvedené atribúty.

Paket môžeme tak definovať ako n-ticu:

$$p = (id, len, src, dst, data), \tag{3.1}$$

kde id, len, sú celé čísla, src a dst sú bitové reťazce konečnej dĺžky a dátová časť aplikačnej vrstvy $data$ je postupnosť bitov. Všetky prvky a n-tice p budeme obecne nazývať **atribúty paketu** a obecne značiť $a \in p$.

Vzhľadom na povahu prenosu informácií po sieti budeme uvažovať, že jednotlivé spojenia neobsahujú chyby, ktoré vznikajú pri prenose paketov sieťovými prvkami (opakujúce sa pakety, chýbajúce pakety, chybné pakety a pod.). Budeme predpokladať, že korekcie na úrovní jednotlivých paketov a spojení sú zaručené nižšou vrstvou (napr. fyzickou). Pre minimálne podmienky na fungovanie takéhoto minimalistického protokolu je potrebné definovať **spojenie** medzi účastníkmi komunikácie (uzlami v sieti) a pre účely analýzy komunikácie bude zavedený pojem **charakteristika spojenia**.

3.2 Definícia spojenia a charakteristiky spojenia

Množinu všetkých paketov, ktoré sú zaznamenané a nad ktorými prebieha následná analýza budeme označovať U. Táto množina môže byť nekonečná, pretože z pohľadu analýzy paketov a práce nad sieťovým tokom je zaznamenávanie paketov nepretržitý proces.

$$U = \{p_1, p_2, \ldots\} \tag{3.2}$$

Nad množinou U je možné definovať podmnožiny paketov, ktoré sú jednoznačne určené dvojicou adries, a to zdrojovou (src) a cieľovou adresou (dst). V spojení budeme predpokladať, že neexistujú dve také spojenia, ktoré majú rovnaké zdrojové a zároveň cieľové adresy

alebo rovnakú cieľovú a zdrojovú adresu a naopak. Definujeme neprázdnú podmnožinu U_C všetkých paketov U:

$$U_C \subseteq U; \quad \forall p_i, p_j \in U_C : \tag{3.3}$$

$$(src(p_i) = src(p_j) \wedge dst(p_i) = dst(p_j)) \vee (src(p_i) = dst(p_j) \wedge dst(p_i) = src(p_j)), \tag{3.4}$$

kde $src(p)$ je zdrojová adresa a $dst(p)$ je cieľová adresa paketu p. Táto podmnožina reprezentuje komunikáciu medzi zdrojom a cieľom a záleží na smere posielaných paketov, či je adresa zdroja (iniciátora spojenia) uvedená ako zdrojová alebo cieľová adresa daného paketu.

Definícia 3.2.1. Definujeme postupnosť C, ktorá je tvorená množinou paketov U_C:

$$C = \{p_1, p_2, \ldots\}, p_i \in U_C, \tag{3.5}$$

kde n-tý člen postupnosti (a tým i poradie všetkých paketov v C) je daný jeho poradovým číslom $id(p_i)$, kde platí

$$\forall p_i, p_j \in U_C, id(p_i) > id(p_j) \vee id(p_i) < id(p_j). \tag{3.6}$$

V rámci postupnosti neexistujú dva pakety, ktoré majú rovnaké poradové číslo: $id(p_i) = id(p_j)$. Túto postupnosť C budeme nazývať **spojenie** a jej vlastnosti sa budú odvíjať od danej komunikácie. Postupnosť môže byť nekonečná a musí byť tvorená aspoň jedným paketom.

V rámci skúmaného spojenia môžeme komunikáciu rozdeliť na dva smery a to *prichádzajúci smer* a *odchádzajúci smer*.

Definícia 3.2.2. Prichádzajúci smer bude označovaná taká komunikácia, ktorá iniciuje dané spojenie. Pre zjednodušenie budeme uvažovať o prvom pakete $p_0 \in C$, a následne všetky pakety p_i, kde zdrojovej adresa $src(p_i)$ sa rovná adrese $src(p_0)$. Prichádzajúci smer bude ďalej označovaný ako podmnožina postupnosti C, ktorá bude tvoriť postupnosť C^{IN} a odchádzajúce spojenie bude označované ako C^{OUT}. V rámci spojenia C sa budú zdrojová a cieľová adresa v jednotlivých paketoch striedať (vymieňať) v závislosti na smere paketu, preto za referenčný paket z ktorého sa odvodí hodnota zdrojovej a cieľovej adresy bude braný počiatočný (prvý) paket spojenia.

Definícia 3.2.3. V rámci práce budeme skúmať atribúty daného spojenia a hľadať vzory správania nevalidnej (anomálnej) komunikácie. Uvažujme o spojení C, potom

$$X_C = (S, V), \tag{3.7}$$

bude značiť **charakteristiku spojenia** C, kde S je množina *skalárnych hodnôt* charakterizujúcich dané spojenie a V je množina, ktorá obsahuje *variabilné hodnoty* charakterizujúce spojenie C (podľa nižšie definovaných pravidiel).

Definícia 3.2.4. Ďalej definujeme **skalárny atribút spojenia**, ktorý je v rámci všetkých paketov daného spojenia rovnaký (nemenný) a nadobúda tak v spojení konštantnú hodnotu. Skalárne atribúty daného spojenia zaradíme do množiny S podľa nasledujúcej definície. Majme množinu A obsahujúcu hodnoty atribútu a pre všetky prvky p_i,

$$A = \{a_i | a_i \in p_i, \forall p_i \in C\}, \tag{3.8}$$

kde a_i je hodnota atribútu a paketu p_i. Atribút a je v rámci spojenia konštantný a je prvkom množiny S práve vtedy, ak platí

$$\forall a_i, a_j \in A : a_i = a_j \rightarrow a \in S. \tag{3.9}$$

Definícia 3.2.5. Ak je hodnota atribútu a pre aspoň jedno $p_i \in C$ rozdielna, budeme hovoriť o **variabilnom atribúte spojenia** a budeme definovať vektor v,

$$\exists a_i, a_j \in A : a_i \neq a_j \rightarrow v = (a_1, a_2, \ldots, a_n), \tag{3.10}$$

kde $v \in V$ a $n = |C|$ je veľkosť spojenia.

Čas je ako skalárny (čas začiatku, čas ukončenia spojenia, trvanie spojenia), tak i variabilný atribút (čas príchodu jednotlivých paketov), ktorý nie je reflektovaný v rámci uvedených pravidiel a nie je ani súčasťou definície paketu minimalistického sieťového protokolu. Čas je ale nezanedbateľný atribút pri skúmaní jednotlivých spojení a to z pohľadu štatistického: čas, v rámci dňa, kedy nastala daná komunikácia; a relatívny čas (čas príchodu jednotlivých paketov) v rámci analýzy prenosu. Čas má i vplyv na vypršanie spojenia (z angl. *time-out*) u niektorých protokolov, služieb a zariadení.

Definícia 3.2.6. Čas ako skalárna hodnota je do charakteristiky spojenia X_C vložený ako atribút času prvého paketu (začiatok spojenia) a posledného paketu (ukončenie spojenia), ktorý je 0-vý v prípade, že spojenie stále trvá (je živé).

Definícia 3.2.7. Čas je do charakteristiky spojenia X_C vložený ako variabilný atribút v_t:

$$v_t = (t_0, t_1, \ldots, t_n),$$

kde t_i je rozdiel času príchodu paketu p_i a p_{i-1}, pre $0 < i \leq |v_t|, p_i \in C$, $v \in V$ a $t_0 = 0$.

Špeciálnym prípadom charakteristiky spojenia (tak isto ako čas, ktorý nie je v atribútoch paketu prítomný) je smer paketov, ktorý je označený dvoma postupnosťami ako C^{IN} a C^{OUT}. Vzhľadom na to, že táto informácia by v charakteristike podľa definícií 3.2.4 až 3.2.7 nebola prítomná, je medzi variabilné vektory pridaný smer ako vektor nominálnych hodnôt.

Definícia 3.2.8. Do charakteristiky spojenia X_C je vložený variabilný atribút v_d, ktorý sa skladá zo zložiek nadobúdajúcich hodnôt 0 alebo 1. Hodnoty označujú smer paketu:

$$v_d = (d_0, d_1, \ldots, d_n), \tag{3.11}$$

kde d_i je smer paketu p_i, $p_i \in C$ a $v_d \in V$ a platí

$$d_i = \begin{cases} 0 & \text{ak } p_i \in C^{OUT} \\ 1 & \text{ak } p_i \in C^{IN} \end{cases} \tag{3.12}$$

Definície 3.2.4 až 3.2.8 vytvorili kompletnú charakteristiku spojenia, ktorá obsahuje všetky atribúty analyzovaných paketov, ktoré sú rozdelené na skalárne atribúty, reprezentované číslom a variabilné atribúty, ktoré sú v charakteristike reprezentované vektorom s hodnotami atribútu daných paketov (hodnota je celé číslo, ale môže byť reprezentované ako postupnosť bitov). Vzhľadom na to, že žiaden atribút paketov neobsahuje informáciu o čase, je čas vložený podľa definícií 3.2.6 a 3.2.7 do charakteristiky ako skalárne hodnoty

začatia a skončenia spojenia (čas príchodu prvého a posledného paketu) a ako vektor rozdielu časov príchodov (pri monitorovaní je to doba zachytenia) paketov. V charakteristike sa taktiež nenachádzala informácia o smere jednotlivých paketov spojenia, preto bol podľa definície 3.2.8 do charakteristiky vložený vektor označujúci smer jednotlivých paketov.

Vytvorená kompletná charakteristika X_C spojenia C obsahuje zložku skalárnych a zložku variabilných charakteristík. Veľkosť jednotlivých vektorových zložiek charakteristík spojení je závislá na počte paketov daného spojenia. Pre analýzu spojení (napr. vstup do klasifikačných algoritmov) je žiadúce, aby každý vstup mal konštantnú dĺžku. Nad charakteristikou sú tak vytvorené metriky – funkcie, ktorých výstupom je konštantná dĺžka číselných hodnôt, ktoré tvoria **signatúru spojenia**, ktorá spĺňa podmienku konštantnej dĺžky.

3.3 Definícia metrík spojenia

Po definovaní procesu vzniku charakteristiky spojenia je potrebné vytvoriť na základe parametrizácie spojenia metriky, podľa ktorých bude prebiehať klasifikácia spojení.

Definícia 3.3.1. Metrika je funkcia, ktorej vstupom je číslo alebo vektor a výstupom je číselná hodnota.

Hodnoty (výstupy) metrík sú vytvárané z atribútov spojenia X_C a to podľa nasledujúcich definícií.

Definícia 3.3.2. Pre každý skalárny atribút $s \in S, S \in X_C$, ktorý je reprezentovaný číslom podľa definície 3.2.4, je výstupom metriky φ číslo reprezentované hodnotou tohoto atribútu, ktorý je konštantný pre všetky pakety spojenia C a patrí do množiny hodnôt M_C:

$$\varphi(s) = m, m \in M_C. \tag{3.13}$$

Definícia 3.3.3. Pre každý variabilný ordinálny atribút $v \in V$, ktorý je reprezentovaný vektorom podľa definície 3.2.5 a 3.2.6, definujeme množinu výstupov funkcií f_i,

$$M_C \supset \{m_i | m_i = f_i(v), v \in V\}, \tag{3.14}$$

Táto množina je tvorená výstupmi funkcií, ktorých vstupnou hodnotou je vektor v a výstupom (hodnotou metriky) je číslo. Príkladom takýchto funkcií sú štatistické funkcie (x_i je zložka vektoru $v : x_i \in v$ a $N = |v|$ je veľkosť vektoru v):

- *minimum* – minimum z hodnôt vektoru;

$$f_{min} = \min_{1 \leq i \leq N} x_i$$

- *maximum* – maximum z hodnôt vektoru;

$$f_{max} = \max_{1 \leq i \leq N} x_i$$

- *medián* – medián hodnôt vektoru;

$$f_{med} = \widetilde{v}$$

- *súčet* – súčet hodnôt vektoru;

$$f_s = \sum_{i=1}^{N} x_i$$

- *priemer* – priemer hodnôt vektoru;

$$f_m = \frac{1}{N} \sum_{i=1}^{N} x_i$$

- *štandardná odchýlka.*

$$f_d = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \Big(x_i - f_m(v)\Big)^2}.$$

Množina hodnôt metrík M_C tvorených týmito funkciami môže byť doplnená ľubovoľnými ďalšími funkciami. Pre nominálne variabilné atribúty ale nemajú uvedené štatistické funkcie zmysel (napríklad pri atribúte *poradové číslo id* nemá zmysel počítať uvedené štatistické funkcie, pretože daný atribút je nominálny) a preto je potrebné zaviesť štatistické funkcie i pre variabilné nominálne atribúty.

Definícia 3.3.4. Pre každý variabilný nominálny atribút, ktorý je reprezentovaný nominálnou hodnotou podľa definícií 3.2.5 a 3.2.8, definujeme množinu vektorov, kde každý vektor reprezentuje prvky s hodnotou c.

$$v_c = \{x_i | x_i = c, x_i \in v\}, \tag{3.15}$$

kde x_i je komponent (zložka) variabilného atribútu reprezentovaného vektorom $v \in V$.

Definícia 3.3.5. Ak sú komponenty x_i vektoru v reprezentované postupnosťou bitov, potom pre každý bit $j \in x_i$, $j = 0, 1, \ldots M$, $M = |x_i|$ definujme komponentný vektor v_j, ktorý je definovaný nasledovne:

$$v_j = (x_0^j, x_1^j, \ldots x_N^j), \forall x_i \in v, N = |v|, \tag{3.16}$$

kde x_i^j je j-tý člen postupnosti bitov v komponente x_i vektoru v. Potom pre každý komponentný vektor v_j je možné definovať metriky $f(v_j)$.

Pre definície 3.3.4 a 3.3.5 definujeme metriku $f(v)$ – štatistickú funkciu počet:

Definícia 3.3.6. Počet je definovaný ako funkcia f_n podľa predpisu

$$f_n = \sum_{i=1}^{N} x_i, \tag{3.17}$$

kde x_i je zložka vstupného vektoru, t.j. vektory v_c podľa 3.3.4 alebo komponentné vektory v_j podľa definície 3.3.5.

Vzhľadom na požiadavku konštantnej dĺžky signatúry spojenia, musí byť výstup uvedenej metriky pre všetky hodnoty definície 3.3.4 úplným výčtom hodnôt. V rámci tejto práce bude metrika *počet* v rámci definície 3.3.4 použitá pre variabilný atribút *smer paketu* 3.2.2 nadobudajúci dve hodnoty.

Dôležitým faktorom pri popise spojenia a následnej analýze je samotný priebeh spojenia. Vzhľadom na to, že každé spojenie môže mať iný počet paketov a zároveň nemôžeme vylúčiť analýzu neukončeného (prípadne nekonečného) spojenia, je potrebné atribúty s variabilnými hodnotami spojenia aproximovať. Aproximácie tak transformujú dvojrozmerný priebeh spojenia na jeden rozmer (napr. koeficienty) a vytvárajú behaviorálnu signatúru spojenia (modelujúcu správanie účastníkov komunikácie v rámci sieťového toku).

Pre aproximáciu priebehu spojenia je možné použiť rôzne aproximačné funkcie. Vo vedeckých prácach sa objavuje využitie polynomiálnej aproximácie na komprimáciu dát [102], prípadne články zaoberajúce sa aproximáciou funkcií intenzity obrazu pre detekciu 3-D tvarov [19] a aproximáciu distribúcie úrovne šedi v LCD paneloch pre detekciu defektných pixelov pomocou polynomiálnej aproximácie polynómom 3. rádu [12]. Aproximačné funkcie majú široké využitie hlavne pri matematických problémoch analýzy dát, pri ktorých je dôležitý výber určitých vlastností na úkor zachovania detailu. V rámci tejto práce boli vybrané metódy polynómu n-tého rádu a Fourierova transformácia, ktoré sú popísané v nasledujúcich podkapitolách. V rámci experimentov s navrhnutými klasifikačnými metódami bola testovaná i aproximácia gaussovými krivkami, ktoré ale nie sú zahrnuté v teoretickom rozbore (pre viac informácií pozri kapitola 5).

Pri aproximáciách je vždy dôležité určiť, či je aproximovaný iba jeden smer spojenia (napr. od zdroja k cieľu) alebo je aproximované celé spojenie (tzn. úplná konverzácia zdroja a cieľu i s odpoveďami). V prvom prípade nevzniká žiadna komplikácia s identifikáciou smeru, ale stráca sa informácia o prípadných odpovediach cieľovej stanice. V druhom prípade je potrebné pri aproximácii rozlíšiť smer paketov. V prípade, že aproximované atribúty sú vždy kladnej hodnoty (napr. veľkosť paketu), opačný smer môže byť reprezentovaný zápornou hodnotou. Hodnoty aproximovaných funkcií sú variabilné atribúty analyzovaného spojenia a to v definičnom obore indexov paketov (index je poradové číslo paketu). V prípade, že by definičný obor bol tvorený časom príchodu paketu na sondu, do aproximačných funkcií by bol zanesený šum spôsobený napr. pomalou odpoveďou zdrojového systému, ktorý paket odoslal, zahltením siete apod. V prípade samotnej detekcie by mohol útočník využiť tento šum na jej obídenie.

Podľa definície 3.3.1 je metrika funkcia, ktorej výstupom je číslo. Túto definíciu je ale potrebné upraviť pri použití aproximačných funkcií, ktorých výstupom je postupnosť čísiel konštantnej dĺžky definovanej zvoleným počtom výstupných koeficientov:

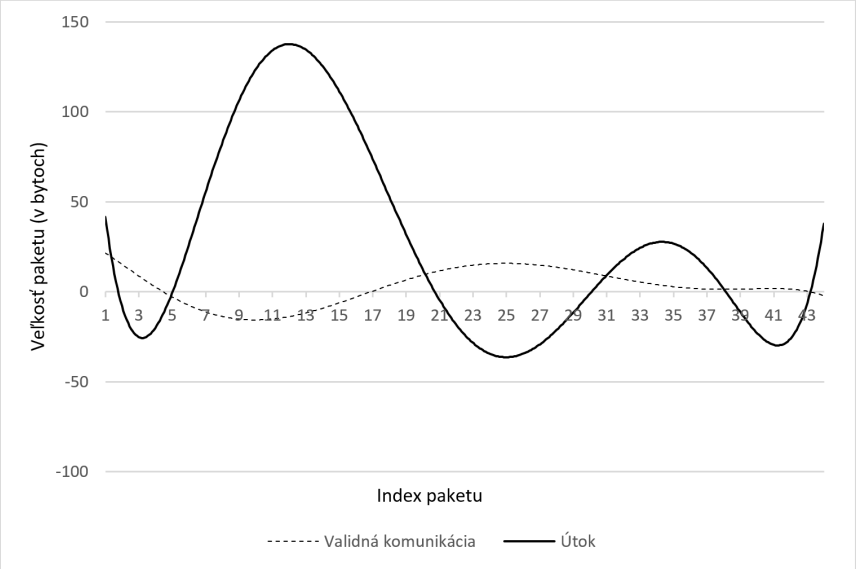
Definícia 3.3.7. Metrika je funkcia, ktorej vstupom je číslo alebo vektor a výstupom je číslo alebo postupnosť čísiel konštantnej dĺžky.

3.3.1 Polynomiálna aproximácia

Aproximácia variabilných atribútov spojenia s dôrazom na veľkú citlivosť funkcie na malé odchýlky v hodnotách vstupného vektoru. Na obrázku 3.1 je zobrazené porovnanie validnej komunikácie a útoku (aproximácie veľkosti paketov v definičnom obore indexov paketov) na ilustrácii fitovania veľkostí paketov pomocou uvedenej polynomiálnej aproximácie (index paketu odpovedá jeho poradovému číslu).

Definícia 3.3.8. Ak je daný atribút a_j variabilným atribútom, potom vstupnými atribútmi $(a_1, a_2, \ldots a_n)$, $a_i \in p$, definujeme funkciu f v indexovej rovine (definičným oborom funkcie je index paketu v spojení) a následne jej aproximáciu $P_n(f)$ polynómom n-tého stupňa. Výstupom je n členov polynómu, ktorý definujeme ako vektor v charakterizujúci spojenie C:

$$v = (p_1, p_2, \ldots, p_n) \tag{3.18}$$



Obr. 3.1: Porovnanie polynomiálnych aproximácií 6. stupňa validnej komunikácie a útoku.

kde p_i je i-tý člen polynómu $P_n(f)$, $v \in X_C$.

Aproximácia polynómom n-tého stupňa môže byť v určitých prípadoch problematická. Hlavnou otázkou pri aproximácii polynómami je stupeň použitého polynómu. V prípade, že spojenie by malo menej paketov ako n, pri polynóme n-tého stupňa dôjde k oscilácii aproximácie a hodnoty polynómu budú nepoužiteľné. Ďalej pri analýze a učení musia byť všetky analyzované spojenia aproximované rovnakým stupňom polynómu, inak by došlo k posunu pri porovnávaní jednotlivých členov polynómov a porovnanie by viedlo k nesprávnym výsledkom. Riešením tejto situácie je použitie predom stanoveného rádu polynómu a v prípade, že počet aproximovaných atribútov spojenia je menej ako rád polynómu, je vstup aproximačnej metódy doplnený nulami.

3.3.2 Fourierova transformácia

Použitie Fourierovej transformácie umožňuje analýzu variabilných atribútov spojenia redukovaných z dvojrozmerného priestoru do jedného rozmeru (vektoru). Riešenie a výsledky analýzy je nasledovne možné preniesť do pôvodnej domény inverznou Fourierovou transformáciou. Aproximácia atribútov spojenia s dôrazom na malú citlivosť funkcie na odchýlky vstupných hodnôt je vhodná pre komprimáciu charakteristiky spojenia.

Definícia 3.3.9. Ak je daný atribút $a \in p$ variabilným atribútom, potom existuje $v \in V, V \in X_C$. Vstupnými atribútmi $(a_1, a_2, \ldots a_n)$, $a_i \in v$, definujeme funkciu f v indexovej rovine (definičným oborom funkcie je index/poradie paketu v spojení a hodnota $f(i)$ je hodnota atribútu a s indexom i) a následne jej aproximáciu $F(f)$ Fourierovou transformáciou. Vzhľadom na to, že vstup je postupnosť paketov, ide o diskrétnu Fourierovu transformáciu (zápis v exponenciálnom tvare):

$$F(n) = \sum_{k=0}^{N-1} f(k) e^{-ink2\pi/N}, n = 0, 1, \ldots N-1, \tag{3.19}$$

kde $F(n)$ je n-tý koeficient Fourierovej transformácie, N je počet vstupných atribútov funkcie f, ktoré sú stanovené pri definícii signatúry a v prípade menších počtov paketov v spojení sú hodnoty $a_i, |v| < i < N$ doplnené 0. Výstupom Fourierovej transformácie je $N/2 + 1$ koeficientov (druhá polovica koeficientov je symetrická k prvej polovici a nie je potrebný ich výpočet):

$$a + bi = F(n), n = (0 \ldots 10), \tag{3.20}$$

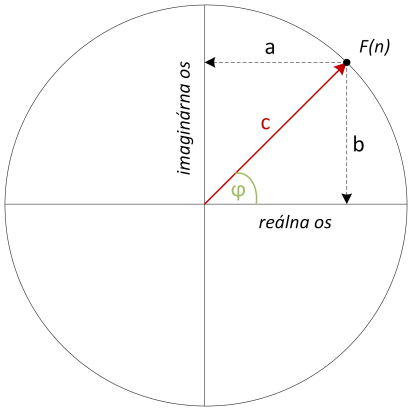
kde a je reálna a b je imaginárna časť koeficientu $F(n)$.

Pri experimentoch bol použitý zápis koeficientov pomocou uhla (φ) a veľkosti vektoru (c) tvoreného reálnou a imaginárnou zložkou koeficientu (pozri obr. 3.2). Tento prevod je možný riešením sústavy rovníc

$$b = a \tan \varphi, \tag{3.21}$$

$$c = \sqrt{a^2 + b^2}, \tag{3.22}$$

kde $(\varphi, c) \in M_C$.



Obr. 3.2: Zobrazenie koeficientu $F(n)$ v komplexnej rovine.

V prípade atribútov aproximačných funkcií je množina hodnôt metrík M_C doplnená o hodnoty metriky, ktoré zodpovedajú jednotlivým členom vektoru danej aproximačnej funkcie. Tzn. v prípade polynomiálnej aproximácii n-tého rádu vznikne jedna hodnota metriky (danej aproximačnej funkcie) na každý člen polynómu:

$$v = (p_1, p_2, \ldots, p_n),$$

kde v_i je i-tý člen polynómu $P_n(f)$. Pre Fourierovu transformáciu sú to dve hodnoty na každý koeficient (zápis veľkosťou a uhlom vektoru). Pre každý variabilný atribút tak vzniknú dve metriky. Napr. pre veľkosť paketu vznikne polynóm n-tého rádu veľkosti paketov a Fourierova transformácia veľkostí paketov v indexovej rovine. Výstupom (hodnotami) metriky polynómu 3-tieho stupňa sú 4 členy tohoto polynómu (a obdobne pre iné prípady). V rámci definície metrík je možné aproximačné funkcie nahradiť inými aproximačnými funkciami, prípadne doplniť napr. o gaussove krivky, spliny, apod.

3.3.3 Dátová časť

Doteraz boli pri vytváraní charakteristiky spojenia a následne metrík používané len hlavičky paketov analyzovaného spojenia. Ďalšou veľmi dôležitou časťou pri klasifikácii sieťových

útokov je dátová časť paketu, ktorá obsahuje dáta vyššieho (budeme uvažovať o aplikačnom) protokolu. V prípade charakteristiky spojenia pomocou hlavičiek paketu je možné rozpoznať aplikačný protokol na základe dvoch príznakov:

- **port služby** – na základe portu alebo cieľovej služby, rezervované, pridelené porty IANA, atď.;
- **atribútom NBAR** – na základe príznaku služby NBAR - *Network Based Application Recognition*.

V oboch prípadoch ide o nepresné informácie, ktoré nemusia zodpovedať definíciám, či rezervovaným portom. Služby môžu bežať na neštandardných portoch (prípadne na portoch rezervovaných iným službám) a v prípade útoku môže útočník zasielať pakety s podstrčeným atribútom *NBAR*.

V prípade, že súčasťou analýzy nebude *DPI* (tzv. *Deep packet inspection*), čiže analýza dátovej časti paketu, je možné zistiť iba niektoré základné vlastnosti prenášaných dát:

- pravdepodobnosť, že dáta sú šifrované (celé číslo v intervale 0 – 100);
- v prípade, že dátová časť nie je šifrovaná, je možné zistiť, či ide o textový alebo binárny protokol rôznymi klasifikačnými metódami, ktoré sú nad rámec tejto práce.

Vzhľadom na informácie získané pri zbieraní dát, je väčšina dnešných protokolov šifrovaná. Vzhľadom na fakt, že šifrované dáta bez príslušnej znalosti kľúča a následnom dešifrovaní obsahu paketov je analýza takýchto dát zbytočná a mohla by znehodnotiť klasifikačný proces, v rámci tejto práce nebude dátová časť paketov z týchto dôvodov súčasťou analýzy.

3.3.4 Zhrnutie

Doteraz bol uvažovaný sieťový tok, ktorý sa skladá z paketov prenášaných medzi dvoma subjektmi na sieti (medzi zdrojom a cieľom). Tento sieťový tok bol pomenovaný **spojenie** a od sieťového toku (z angl. *network traffic flow*, definovaného v RFC 2722 [27] a RFC 3697 [110]) sa odlišuje tak, že obsahuje pakety v oboch smeroch, na rozdiel od toku, ktorý obsahuje pakety vždy iba od zdroja k cieľu, takže vzájomná obojstranná neprázdna komunikácia obsahuje vždy dva toky (v prípade, že cieľ odpovedá a konverzácia je úplná). Nad spojením bola vytvorená **charakteristika spojenia**, ktorá je tvorená množinou skalárnych atribútov a množinou variabilných atribútov (reprezentovaných vektorom). Skalárna časť charakteristiky spojenia je tvorená takými atribútmi paketov (napr. zdrojová adresa paketu), ktoré sú konštantné v rámci celej komunikácie. Variabilná časť je tvorená množinou vektorov, kde každý vektor odpovedá jednému atribútu paketu, ktorý sa v rámci spojenia (v rámci všetkých paketov) mení (napr. veľkosť paketu). Charakteristika spojenia tak popisuje vlastnosti daného spojenia. Nad vlastnosťami, resp. charakteristikou každého spojenia je podľa definícií metrík spojenia (definície 3.3.2 až 3.3.9) vytvorená sada metrík a ich výstupov. **Metrika** je funkcia, ktorej výstupom je číslo alebo postupnosť čísiel konštantnej dĺžky, ktoré reprezentujú jednu vlastnosť tohoto spojenia. Sada týchto metrík pre dané spojenie vytvára vektor hodnôt metrík (tiež pri klasifikačných úlohách často označované ako rysy, z angl. *features*) a tie tvoria tzv. **signatúru spojenia**. Signatúra spojenia je navrhnutá tak, aby čo najlepšie popisovala vlastnosti spojenia, ideálne zachovala maximálne množstvo informácie, pri čo najmenšej veľkosti.

Pre ďalšie použitie metrík definovaných nad minimalistickým sieťových paketovým protokolom je potrebné tento protokol upraviť tak, aby zodpovedal bežne používaným technológiám sieťových technológií založených na paketoch. Definovaný protokol bude v nasledujúcej kapitole rozšírený o vlastnosti TCP/IP architektúry.

3.4 Rozšírenie protokolu na TCP/IPv4

V rámci TCP/IP budeme o adrese uvažovať ako o dvojici IP adresa a port:

$$A = (ip, port) \tag{3.23}$$

V prípade, že budeme hovoriť o IP adresách, budeme atribúty IP adries značiť indexom IP a to src_{IP} pre IP adresu zdroja a dst_{IP} pre adresu cieľa. V tejto kapitole bol definovaný minimalistický sieťový prúdový paketový protokol, ktorý bol následne použitý pri definícii sieťového spojenia, ktoré je reprezentované postupnosťou paketov medzi zdrojom a cieľom. Nad týmto spojením bola definovaná jeho charakteristika, ktorá je daná množinou skalárnych a variabilných atribútov paketov tohto spojenia. Skalárne atribúty sú také, ktoré majú v rámci všetkých paketov spojenia konštantné hodnoty a u variabilných atribútov sa tieto hodnoty v rámci spojenia menia. Skalárne atribúty spojenia v protokole IP u TCP/IP architektúry vo verzii IPv4 môžu byť napríklad:

- **Verzia** – verzia protokolu IP, ktorá je uvedená v hlavičke IP protokolu, v rámci daného spojenia je očakávané celé číslo s konštantnou hodnotou.
- **DSCP** – typ služby, napríklad VoIP, reprezentované číslom (identifikátorom) typu služby, v rámci spojenia by sa identifikácia služby nemala meniť.
- **Protokol** – číselná identifikácia protokolu dátovej časti IP protokolu (v tomto prípade TCP protokol je identifikovaný číslom 6) v rámci jedného spojenia konštantná.
- **Zdrojová a cieľová adresa** – bitové reťazce konštantnej dĺžky identifikujú danú komunikáciu a sú teda v rámci spojenia nemenné.

Ďalej z pohľadu manažmentu prenosu a kontroly zahltenia siete je nutné uvažovať i napr. o fragmentácii paketov (rozdelenie veľkých paketov do série menších paketov) a do atribútov pridať nasledujúce atribúty:

- **Životnosť paketu** – celé číslo, ktoré označuje životnosť paketu, môže ísť napríklad o číslo, ktoré reprezentuje časovú hodnotu, ktorá je neustále znižovaná a po jej uplynutí (dosiahnuté 0) je paket zahodený (RTO podľa RFC 793 [105]);.
- **Ofset fragmentu** – číslo označujúce ofset fragmentu od počiatku pôvodného nefragmentovaného paketu v prípade potrebnej fragmentácie.

V ďalšej vrstve architektúry TCP/IP a to konkrétne TCP protokolu sú príklady skalárnych atribútov: *zdrojový port, cieľový port, rezervované bity* (ktoré sa nepoužívajú), atď.

Pre variabilné atribúty spojenia v uvedenom protokole je možné uviesť nasledujúce príklady atribútov hlavičky IP paketu:

- **Dĺžka** (veľkosť) paketu – počet bitov daného paketu vrátane dátovej časti. Každý paket môže mať rozdielnu veľkosť, ktorá je daná ako premenlivým počtom nepovinných atribútov hlavičky jednotlivých protokolov, tak i veľkosťou dátovej časti.

- **Atribúty** – trojbitová hodnota určujúca fragmentáciu IP paketu na viacero IP paketov (v prípade veľkých dát, ktoré musia byť rozdelené na menšie časti tak, aby spĺňali podmienky danej siete (napr. MTU limit pre veľkosť paketu).

V ďalšej vrstve architektúry TCP/IP a to konkrétne TCP protokolu sú príklady variabilných atribútov: *sekvenčné číslo, potvrdzovacie (acknowledgment) číslo, ofset dát, TCP atribúty (flags)*, atď.

3.4.1 Signatúra spojenia v TCP/IP protokoloch

V predchádzajúcom texte bolo vysvetlené rozšírenie definovaného minimalistického sieťového protokolu na moderný sieťový protokol TCP/IP. Toto rozšírenie je vhodné ako pre IPv4 tak i IPv6 verzie protokolu. V rámci práce, ako už bolo spomenuté, je predmetom analýzy a detekčného systému iba protokol TCP. Po aplikovaní pravidiel z definícií 3.2.4 až 3.2.7 nad TCP/IP protokolom verzie 4 dostaneme pre analyzované spojenie nasledujúcu charakteristiku spojenia:

- **Skalárne atribúty protokolu** (definícia 3.2.4): verzia protokolu IP, DSCP, ECN, identifikácia skupín fragmentov, Time to Live, protokol, zdrojová adresa, cieľová IP adresa, zdrojový port, cieľový port.
- **Skalárne časové atribúty** (definícia 3.2.6): čas príchodu prvého paketu, čas príchodu posledného paketu.
- **Variabilné atribúty** (definícia 3.2.5): IHL, dĺžka IP paketu, IP atribúty, ofset fragmentu, kontrolný súčet hlavičky IP, IP rozšírené možnosti – Options, Sekvenčné číslo, číslo potvrdenia, potvrdzovacie číslo, ofset dát, rezervované bity, TCP atribúty, veľkosť okna, kontrolný súčet hlavičky TCP, urgent pointer, TCP rozšírené možnosti – Options.
- **Variabilný časový atribút** (definícia 3.2.7): časový rozdiel príchodu paketu od príchodu predchádzajúceho paketu.
- **Smer paketu** (definícia 3.2.2): nominálna hodnota určujúca smer paketu.

3.4.2 Metriky v TCP/IP

Vzhľadom na veľké množstvo atribútov vytvorených z TCP/IP protokolov, je potrebné tieto dve množiny skalárnych a variabilných atribútov zmenšiť. Vzhľadom na povahu informácií, ktoré jednotlivé atribúty nesú je logické niektoré z nich odstrániť už pri návrhu tohoto systému. Sú to hlavne atribúty, ktoré sú za každých okolností konštantné, *konkrétne verzia protokolu IP (konštantná hodnota pre protokol IPv4), protokol (bude konštantný s hodnotou 6 – TCP), Rezervované bity (vždy 0).*

Ďalej sa medzi variabilnými atribútmi nachádzajú také, ktoré z určitého pohľadu nedávajú pri analýze zmysel a to z dôvodu ich účelu pre potreby riadenia TCP alebo IP protokolu ako napr. kontrolné súčty a pod. Sú to konkrétne: *kontrolný súčet hlavičky IP* a *hlavičky TCP*, *ofset fragmentu* a *ofset dát*, *IP atribúty* a *IP identifikácia.* Uvedené atribúty okrem kontrolných súčtov sa týkajú IP fragmentácie, ktorá v prípade analýzy môže spôsobovať problémy (informácia o veľkosti dát je sploštená do maximálnej veľkosti MTU) a preto pred analýzou a vytvorením signatúry spojenia je potrebné vykonať defragmentáciu paketov a to i pre IP fragmentáciu aj pre TCP segmentáciu. Po defragmentácii sú

tieto informácie zbytočné a z atribútov pre vytvorenie signatúry sú zahodené. S hodnotami Options u IP i TCP protokole sa počíta ako s ordinálnymi hodnotami.

3.4.3 Defragmentácia (Reassembling)

Sieťový protokol 2 vrstvy (IP) podporuje tzv. fragmentáciu datagramov, tzn. všetky pakety, ktoré sú väčšie ako definované MTU (v rámci danej siete), sú fragmentované na menšie pakety. Defragmentácia (tzv. reassembling) paketov je veľmi jednoduchý algoritmus, ktorý je inverzný k tomuto procesu. Tento istý proces je i na TCP protokole.

3.4.4 Výpočet signatúry spojenia

Spracovaná charakteristika spojenia obsahuje 6 skalárnych atribútov TCP/IP protokolu a 10 variabilných parametrov.

Výstupom metrík definovaných v 3.3.3 sú pre každý variabilný atribút spočítané funkcie minimum, maximum, medián, súčet, priemer a štandardná odchýlka. Podľa definícií 3.3.8 a 3.3.9 sú pre každý variabilný atribút spočítané funkcie pre aproximáciu atribútu (vektoru hodnôt) polynómom n-tého stupňa a to pre prichádzajúcu, odchádzajúcu komunikáciu a oba smery a Fourierovou transformáciou pre oba smery. V prípade napríklad polynómu 3-tieho a 5-teho stupňa, a uchovávanými 10-timi koeficientmi Fourierovej transformácie je pridaných (polynóm n-tého stupňa má n+1 koeficientov) 56 hodnôt do signatúry na jeden variabilný atribút, čo celkovo činí **593 hodnôt** pre TCP/IP protokol verzie 4. V prípade použitia vyššieho stupňa polynómu alebo viac vypočítaných koeficientoch Fourierovej transformácie, je možné dostať viac výstupov metrík (je tým možné ovplyvniť presnosť aproximácie a tým i úspešnosť klasifikácie), ale pri zvýšenom nároku na výpočtový výkon. Prehľad použitých atribútov TCP/IP protokolov je uvedený v tabuľke 3.1.

V prípade, že sú do sady metrík pridané hodnoty smeru paketov (podľa 3.2.2) a čas (podľa 3.2.6 a 3.2.7), dostaneme 6 hodnôt pre smer (počet zložiek podľa 3.3.6), a 58 hodnôt pre čas, celkový počet hodnôt základných metrík je tak **657**. I v prípade, že variabilný atribút spojenia má vo všetkých paketoch konštantnú hodnotu, v rámci výpočtov a výslednej signatúry bude braný stále ako variabilný atribút a to z dôvodu zachovania formátu signatúry spojenia pre jej ďalšiu analýzu.

Veľkosť signatúry. Signatúra jedného analyzovaného spojenia v prípade uvedených metrík a použitím polynómu 3-tieho a 5-teho stupňa a 10 koeficientov Fourierovej transformácie je zložená z nasledujúcich častí:

- **256** bitov skalárnych atribútov;
- **2640** bitov na štatistické metriky variabilných atribútov (3.3.3 – 3.3.5);
- **2112** bitov pre polynomiálnu aproximáciu (3. rádu) príchodzieho, odchodzieho a oboch smerov;
- **3168** bitov pre polynomiálnu aproximáciu (5. rádu) príchodzieho, odchodzieho a oboch smerov;
- **7040** bitov pre Fourierovu transformáciu oboch smerov.

Celková teoretická veľkosť signatúry je cca 1902 bytov na spojenie (veľkosť sa môže líšiť od implementácie). V prípade analýzy veľkých dátových tokov je možné prispôsobiť signatúry voľbou vhodnejších metrík, napr. vynechať aproximáciu Fourierovou transformáciou, čím sa ušetrí až 46 % (880 bytov na spojenie) celkovej veľkosti.

Atribút	**Protokol**	**Typ**	**Veľkosť**	**Počet metrík**
Version	IPv4	-	-	-
IHL	IPv4	variabilný	4 b	56
DSCP	IPv4	skalárny	6 b	1
ECN	IPv4	skalárny	2 b	1
Total Length	IPv4	variabilný	16 b	56
Identification	IPv4	-	-	-
Flags	IPv4	-	-	-
Fragment Offset	IPv4	-	-	-
TTL	IPv4	variabilný	8 b	56
Protocol	IPv4	-	-	-
Header Checksum	IPv4	-	-	-
Source address	IPv4	skalárny	32 b	1
Destination address	IPv4	skalárny	32 b	1
Options	IPv4	variabilný	32 b	56
Source port	TCP	skalárny	16 b	1
Destination port	TCP	skalárny	16 b	1
Sequence number	TCP	variabilný	16 b	56
Acknowledgment number	TCP	variabilný	32 b	56
Data Offset	TCP	-	-	-
Reserved	TCP	-	-	-
Flags	TCP	variabilný	9 b	83
Window size	TCP	variabilný	16 b	56
Checksum	TCP	-	-	-
Urgent pointer	TCP	variabilný	16 b	56
Options	TCP	variabilný	0 - 320 b	56
Suma	.	.	.	**593**

Tabuľka 3.1: Prehľad atribútov TCP/IP architektúry (pre verziu IPv4 protokolu).

3.4.5 TCP/IP verzie 6

Nová verzia špecifikácie IP protokolu (IPv6) nesie niekoľko zmien oproti IPv4. Hlavička IP paketu má konštantnú dĺžku (nie je možnosť pridávať parametre atribútu *Options*), ale pribudla možnosť pridávať za hlavičku ďalšie hlavičky iných protokolov. Z IPv6 špecifikácie tak vypadli položky *IHL*, *Identification*, *Flags*, *Fragment Offset*, *Header Checksum*, *Options* a *Padding*. Do IPv6 pribudla 20 bitová hodnota *Flow Label*, ktorá mala pôvodne slúžiť na identifikáciu tokov, u ktorých je dodržovaný routing tak, aby nedochádzalo k narušeniu poradia prichádzajúcich paketov [111]. Ostatné položky ostali rovnaké a vytvorenie signatúry pre IPv6 protokol je analogické.

3.5 Kontext spojenia

Doteraz boli pri vytváraní sieťových metrík používané len hlavičky TCP a IP paketov konkrétneho analyzovaného spojenia. Ďalšou veľmi dôležitou časťou pri klasifikácii sieťových útokov sú informácie o analyzovanom spojení vzhľadom na iné prebiehajúce spojenia. Analytické metódy postavené nad sieťovým tokom väčšinou používajú tzv. okno spojenia (pozri literatúru z kapitoly 2), ktoré je definované ako časový úsek od začatia spojenia trvajúci definovaný čas. Tento prístup má niekoľko slabín. V prvom rade sú všetky analyzované spojenia klasifikované až po uplynutí doby definovaného týmto oknom (čo v prípade bezpečnostných systémov býva spravidla viac ako 3 minúty z dôvodu vypršania času TCP spojenia). Tento čas je príliš dlhá doba na to, aby analytický systém dokázal zareagovať na zistenú hrozbu. Druhý zásadný problém pri použití časových okien vytvárajú spojenia, ktorým vyprší povolený definovaný čas presunu TCP paketov (tzv. *TCP retransmision time-out*) [4]. V prípade útokov na sieťové služby pomocou vektoru buffer overflow je častým sprievodným úkazom útoku vypršanie pôvodného spojenia z dôvodu pádu obsluhujúceho procesu (na chybu *segmentation fault*). Následne spojenie vyprší po niekoľkých sekundách, až minútach a tak pri analýze je nutné čakať na všetky spojenia, ktoré nie sú validne ukončené. V extrémnej situácii môže tento stav spôsobiť až vyčerpanie zdrojov systému pri analýze. Táto situácia je ešte horšia v prípade analýzy viacerých prebiehajúcich spojení, kde sa okno musí udržať pre všetky simultánne bežiace spojenia, ktoré začali počas analyzovaného spojenia až do ich ukončenia. Zložitosť sa tak z lineárnej (každé spojenie je analyzované zvlášť) mení na exponenciálnu (každé spojenie je analyzované vzhľadom ku všetkým ostatným).

Definícia 3.5.1. Ďalej zavedieme pojem **kontext** K **spojenia** C, ktorý je definovaný ako množina spojení:

$$K = \{s; wtime(s) = wtime(C), s \neq C\} \tag{3.24}$$

kde $wtime(s)$ je aktuálne časové okno daného spojenia a s je signatúra spojenia. V niektorých prípadoch analýzy aktuálneho spojenia je potrebné analyzovať informácie spojené s aktérmi, ktorí vystupujú v rámci analyzovaného spojenia, t.j. zdrojový a cieľový uzol siete.

Definícia 3.5.2. Definujeme **lokálny kontext** K_L **spojenia** C:

$$K_L = \{s; src_{IP} \in add(s) \wedge dst_{IP} \in add(s)\}, src_{IP}, dst_{IP} \in C, \tag{3.25}$$

kde $add(s)$ je množina adries spojenia (konkrétne zdroj a cieľ), src_{IP} je zdrojová adresa a dst_{IP} je cieľová adresa spojenia C. Lokálny kontext tak obsahuje všetky signatúry spojení medzi aktérmi analyzovaného spojenia. Pri analýze IPv4 protokolu sa jedná iba o IP adresy aktérov bez portov, pretože porty (minimálne zdrojový) sú vo väčšine prípadov v spojení generované dynamicky.

Definícia 3.5.3. Globálny kontext K_G **spojenia** C je rozšírenie lokálneho kontextu o všetky aktuálne signatúry spojení oboch aktérov:

$$K_G = \{s; src_{IP} \in add(s) \vee dst_{IP} \in add(s)\}, src_{IP}, dst_{IP} \in C. \tag{3.26}$$

Kontext spojenia je možné chápať ako informáciu o aktuálnom stave prostredia spojenom s daným analyzovaným spojením. V prípade, že nás bude zaujímať iba dvojica aktérov, ktorí vystupujú v rámci danej komunikácie, teda zdroj a cieľ komunikácie, budeme hovoriť o **lokálnom kontexte**.

Vzhľadom na to, že v čase je možné kontext neustále udržovať, môže byť z definície 3.5.1 vypustená obmedzujúca podmienka časového okna a dostaneme úplnú históriu komunikácie aktérov analyzovaného spojenia (v prípade globálneho kontextu tak všetky komunikácie oboch aktérov) a lokálny kontext C môžeme definovať ako:

$$K = \{s; s \neq C \wedge src_{IP} \in add(s) \wedge dst_{IP} \in add(s)\} \tag{3.27}$$

Pre globálny kontext je definícia adekvátna definícii 3.5.3. Obmedzujúcou podmienkou časového okna v rámci ktorého uvažujeme o kontexte spojenia je možné ovplyvňovať nároky systému na výkon.

Príklad: *V rámci analýzy zachyteného spojenia budeme skúmať predchádzajúce komunikácie oboch aktérov a v prípade, že v historických záznamoch zdroj nikdy nekomunikoval na cieľový systém, budeme komunikáciu považovať za podozrivú*[1]*. Ak je k dispozícii dostatočný výkon, je možné dynamicky rozširovať časové okno analyzovaného spojenia (historických dát, kde pri zvyšujúcom sa časovom okne globálneho kontextu narastá i počet analyzovaných systémov). Pri obmedzenom výkone je možné toto okno zmenšovať, čím dochádza k znižovaniu presnosti detekcie.*

3.6 Metódy analýzy spojení

V predchádzajúcom texte bol definovaný proces spracovania sieťového toku, ktorý podľa definovanej množiny metrík je redukovaný na vektor hodnôt, tzv. signatúru spojenia. Táto signatúra slúži ako vstupná množina hodnôt pre klasifikačné algoritmy, ktoré majú za cieľ identifikovať v danej komunikácii prebiehajúci útok a to na základe testovacej množiny signatúr.

3.6.1 Extrakcia rysov

Celé spojenie bolo v popisovanom procese redukované na signatúru a táto signatúra vystupuje v ďalšom procese vyhodnocovania a analýzy (nebudeme ďalej uvažovať o spojení, ale iba o signatúre spojenia). Ďalšou vrstvou zapojenou do tohoto procesu je vrstva, ktorá sa skladá z popísaných metód analýzy. Vstup do každej metódy je signatúra analyzovaného spojenia. Vzhľadom na množstvo informácií, ktoré nesie signatúra je potrebné vybrať iba konkrétne rysy (komponenty) signatúry pre vstup do klasifikačných algoritmov. Redukcia rysov je nutná pre zníženie celkového počtu dimenzií pri klasifikácii vstupných dát a to z pohľadu komplikovanosti modelov, zníženiu potrebného výkonu, prostriedkov a času pre učiaci proces. Problémom pri výbere jednotlivých rysov je zložitosť procesu výberu ich podmnožín (prehľadanie celého priestoru nie je časovo možné) a najlepšie výsledky môže prinášať kombinácia niekoľkých rysov.

Analýza základných komponent (PCA)

Analýza základných komponent (z angl. *Principal Component Analysis*) je metóda používaná na redukovanie dimenzií vstupných dát. Použitie PCA je vhodné v prípade, že vstupné dáta obsahujú veľký počet atribútov, ktoré obsahujú redundantnú informáciu a nájdením

[1] Podozrivá komunikácia nemusí byť označená za útok, ale môže byť napr. podrobená hlbšej analýze (záleží na konkrétnej implementácii a nastavení detekčného systému).

vzájomných korelácií medzi jednotlivým atribútmi je možné počet atribútov redukovať (vytvorením množiny novo-vytvorených atribútov) na najrozdielnejšie atribúty (atribúty s najväčšou vzájomnou odchýlkou hodnôt).

Princípom PCA je vyhľadanie tzv. vlastných vektorov a príslušných vlastných hodnôt (pre viac informácií pozri [70]). Majme definíciu štatistickej funkcie rozptyl, ktorá je vyjadrením variability hodnôt okolo priemeru:

$$S^2 = \frac{\sum_{i=1}^{m} (x_i - \overline{X})^2}{(m-1)}, \tag{3.28}$$

kde m je veľkosť množiny hodnôt X, $\overline{X}$ je priemer hodnôt množiny X a $x_i \in X$.

Rozptyl je používaný pri štatistickej analýze súboru dát v rámci jednej dimenzie (jeden atribút). V prípade, že je potrebné analyzovať dát so zameraním na nájdenie korelácie medzi vstupnými atribútmi (počet atribútov je počet dimenzií), používa sa kovariancia (z angl. *covariance*), ktorá je vyjadrením vzťahu dvoch atribútov:

$$C = \frac{\sum_{i=1}^{m} (x_i - \overline{X})(y_i - \overline{Y})}{(m-1)}. \tag{3.29}$$

Pri viac ako dvoch porovnávaných atribútoch (napr. n atribútoch) je vypočítaná štvorcová matica $\mathbf{A}^{n \times n}$, kde prvok $x_{i,j}$ je kovariancia medzi i-tým a j-tým atribútom vstupných dát. Vektor $\tilde{\mathbf{v}}$ je vlastný vektor matice $\mathbf{A}$ práve vtedy, ak existuje také číslo λ, že platí:

$$\mathbf{A}.\tilde{\mathbf{v}} = \lambda \tilde{\mathbf{v}}. \tag{3.30}$$

Algoritmus PCA aplikuje popísané základné informácie o vlastných vektoroch a hodnotách pre redukciu dimenzií atribútov vstupnej dátovej sady a je možné tento algoritmus stručne popísať nasledujúcim postupom:

1. Vypočítanie priemeru pre všetky atribúty (dimenzie) vstupných dát (potrebný pre výpočet rozptylu v 3.28) a jeho odčítanie od hodnôt dátovej sady (pre vycentrovanie všetkých hodnôt okolo počiatku súradnicovej sústavy).

2. Spočítanie matice kovariancií ako je definované v 3.29;

3. Výpočet vlastných vektorov a vlastných čísiel podľa 3.30;

4. Výber vhodných vlastných vektorov podľa veľkosti vlastných hodnôt (najvyššia je najlepšia) a vynechanie vektorov s najmenšími vlastnými hodnotami (pre redukciu dimenzií);

5. Dáta sú prevedené do novej súradnicovej sústavy podľa vybraných vlastných vektorov (vektory sú navzájom cez všetky dimenzie ortogonálne);

Vzhľadom na to, že atribúty signatúry TCP/IPv4 sieťovej komunikácie obsahujú veľa redundantnej informácie, minimalizácia tejto redundancie by mohla priniesť zmenšenie veľkosti signatúry a tým zvýšiť výkon systému a mieru úspešnosti klasifikácie, vedľa primárnej funkcie, ktorou je transformácia existujúcich N-dimenzionálnych dát (N je počet atribútov) na menší počet nových, kvalitnejších atribútov.

Forward Selection (FS)

Forward selection (alebo aj *forward feature selection*, FFS, je možné preložiť ako dopredný výber rysov, ale preklad by nebol presný) je metóda výberu rysov na základe testovacej podmienky (napr. úspešnosti klasifikácie pomocou zvoleného klasifikačného algoritmu, regresie alebo iným zvoleným kritériom). Na rozdiel od PCA, táto metóda neredukuje dimenzie rysov, ale vyberá tie, ktoré majú najlepšie výsledky zvolenej testovacej podmienky. Algoritmus funguje na základe pridávania nových rysov na základe jeho ohodnotenia vzhľadom na množinu už vybraných rysov, vždy začína s prázdnou množinou a pridáva jeden atribút v rámci jedného kola výberu. Forward selection býva často označovaný i ako *sequential forward selection* (SFS) [140]. Ďalej je možné spomenúť algoritmus *sequential backward selection* (SBS), ktorý na rozdiel od SFS začína s plnou množinou rysov a odstraňuje najmenej vhodné rysy pomocou definovanej podmienky [86]. Obe základné metódy majú svoje limity pri výbere optimálnej množiny rysov [108] a tieto limity stáli za vznikom ďalších metód ako *sequential forward floating selection* (SFFS), *sequential backward floating selection* (SBFS), ale i ďalšie. Pre účely tejto práce bude uvedený základný princíp použitého SFFS algoritmu [86]:

Majme celkový počet n rysov, prvkov množiny Y a množina $X = \{x_i\}, |X| = k$ je podmnožinou množiny Y. Funkcia $J(x_i)$ je kritérium výberu prvku (testovacia podmienka). Potom definujeme *najvýznačnejší prvok* f_i z množiny $Y - X$:

$$Y - X = \{f_i; i = 1, 2, \ldots, n - k; f_i \in Y, f_i \neq x_j; \forall x_i \in X\}, \tag{3.31}$$

potom množina $Y - X$ je množina voľných rysov k výberu a X_k je vybraná množina najvýznamnejších prvkov podľa kritéria výberu J. Definujeme *individuálny význam* rysu $S_0(x_i)$ vzhľadom na množinu vybraných rysov X_k ako:

$$S_{k-1}(x_i) = J(X_k) - J(X_k - x_i) \tag{3.32}$$

a význam S_{k+1} vybraného $k + 1$ prvku x_{k+1} vzhľadom na množinu X_k, $k = |X|$, ako

$$S_{k+1}(f) = J(X_k + f) - J(X_k). \tag{3.33}$$

Potom je možné povedať, že prvok x_i je najvýznamnejší prvok voči množine X_k práve vtedy, ak

$$S_{k+1}(f_j) = \max_{1 \leq i \leq n-k} S_{k+1}(f_i). \tag{3.34}$$

Adekvátne najmenej významný prvok množiny X_k je

$$S_{k-1}(x_j) = \min_{1 \leq i \leq k} S_{k-1}(x_i). \tag{3.35}$$

- **Krok 1 (inklúzia)**. Výber rysu x_i (*najvýznačnejšieho prvku*) z množiny voľných rysov $Y - X$ a vytvorenie novej množiny $X_{k+1} = X_k + x_i$.
- **Krok 2 (podmienená exklúzia)**. Nájdenie najmenej významného prvku množiny X_{k+1}. Ak je najmenej významný prvok práve pridaný prvok x_i, priraď $k = k + 1$ a pokračuj krokom 1. Ak $x_r, 1 \leq r \leq k$ je najmenej významný prvok, potom vylúč prvok x_r z X_{k+1}:
$$X_k^{'} = X_{k+1} - x_r.$$

Ak $k = 2$, potom nastav $X_k = X_k^{'}$ a pokračuj krokom 1. Inak pokračuj krokom 3.

- **Krok 3 (pokračovanie podmienenej exklúzie)**. Nájdenie najmenej významného prvku x_s množiny $X_k^{'}$. Ak $J(X_k^{'} - x_s) \leq J(X_{k-1})$, potom priraď $X_k = X_k^{'}$ a pokračuj krokom 1. Inak vylúč x_s z $X_k^{'}$ pre vytvorenie redukovanej množiny $X_{k-1}^{'}$:

$$X_{k-1}^{'} = X_k^{'} - x_s.$$

Nastav $k = k - 1$. Ak $k = 2$, potom priraď $X_k = X_k^{'}$ a $J(X_k = J(X_k^{'})$ a pokračuj krokom 1, inak opakuj krok 3.

Algoritmus je inicializovaný priradením $k = 0$ a $X_0 = \emptyset$ a SFS metóda je používaná kým výsledná množina rysov nemá mohutnosť 2, potom algoritmus pokračuje krokom 1. Metóda SFFS je v podstate pridávanie nových rysov do existujúcej množiny pomocou klasického SFS algoritmu s pridaním krokov podmienenej exklúzie najhorších rysov, čím je dosiahnuté zlepšenie pôvodného algoritmu [86].

3.6.2 Klasifikačné algoritmy

Pre základnú klasifikáciu analyzovanej sieťovej komunikácie je možné použiť rôzne matematické metódy, algoritmy strojového učenia apod. Pre túto prácu boli vybrané niektoré metódy nad ktorými boli vytvorené experimenty. Tieto metódy vychádzajú z aktuálnych vedeckých prác v tejto oblasti (pozri kapitolu 2).

Bayesovské klasifikátory

Bayesovské klasifikátory sú štatistické klasifikátory, ktoré dokážu určiť pravdepodobnosť príslušnosti do tried, napr. že daná vzorka dát patrí do určitej triedy. Bayesovské klasifikátory sú založené na Bayesovej vete (Bayes theorem) [72, 60].

Nech T je množina trénovacích vzoriek, ktoré sú zaradené do n tried $C_1, C_2, \ldots, C_n$. Každá vzorka $X \in T$, je reprezentovaná n-dimenzionálnym vektorom $X = x_1, x_2, \ldots, x_n$ reprezentujúcim hodnoty n atribútov $A_1, A_2, \ldots, A_n$. Vstupom bayesovského klasifikátora je vzorka X a výstupom je pravdepodobnosť s ktorou vzorka X patrí do triedy C_i. Táto pravdepodobnosť je uvádzaná v literatúre [72, 60] ako *„a posteriori“*, čo značí, že ide o pravdepodobnosť určenú za vstupných podmienok, napr. $P(A|B)$ je pravdepodobnosť javu A za predpokladu javu B.

Vzorka X patrí do triedy C_i práve vtedy, ak

$$P(C_i|X) > P(C_j|X) \tag{3.36}$$

pre $1 \leq j \leq m, j \neq i$, pre m klasifikačných tried. Z uvedeného vzorca vyplýva hľadaná maximálna hodnota $P(C_i|X)$. Trieda C_i, pre ktorú je hľadaná maximálna hodnota $P(C_i|X)$ sa nazýva maximálna posteriori hypotéza. Podľa Bayesovho teorému

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{3.37}$$

Vzhľadom na to, že $P(X)$ je pre všetky triedy rovnaký, je potreba maximalizovať iba $P(X|C_i)P(C_i)$, Ak nepoznáme hodnotu a priori pravdepodobnosti $P(Ci)$, tak budeme uvažovať, že pravdepodobnosti pre všetky triedy sú rovnaké: $P(C_1) = P(C_2) = \ldots = P(C_k)$

a budeme hľadať maximum pre $P(X|C_i)$, v opačnom prípade budeme hľadať maximum $P(X|C_i)P(C_i)$. Hodnotu a priori pravdepodobnosti triedy $P(C_i)$ je možné určiť z trénovacích vzoriek pomocou

$$P(C_i) = \frac{freq(C_i, T)}{|T|} \tag{3.38}$$

Vzhľadom na výpočtovú náročnosť spočítania jednotlivých pravdepodobností pre všetky triedy a kombinácie všetkých parametrov $P(X|C_i)$ budeme pristupovať k výpočtom pravdepodobnosti tzv. naivným predpokladom podmienenej nezávislosti, čo znižuje celkovú náročnosť výpočtu kombinácií pravdepodobnosti iba na vzťah:

$$P(C_i|X) \approx \prod_{k=1}^{n} P(x_k|C_i), \tag{3.39}$$

kde n je počet atribútov vo vzorke X. Jednotlivé pravdepodobnosti

$$P(x_1|C_i), P(x_2|C_i), \ldots, P(x_n|C_i)$$

sú určené z trénovacej množiny. V prípade, že atribút A_k môže nadobúdať hodnoty z predom stanovenej množiny hodnôt (z angl. *categorical value*, napr. môže nadobúdať hodnotu pravda alebo nepravda), pravdepodobnosť $P(x_k|C_i)$ je daná pomerom počtu vzoriek x_k, ktoré sa nachádzajú v triede C_i v trénovacej množine T a celkového počtu kategórií C_i v trénovacej množine:

$$P(x_k|C_i) = \frac{freq^x(C_i, T)}{freq(C_i, T)}, \tag{3.40}$$

kde $freq^x(C_i, T)$ je počet tried C_i, ktoré obsahujú vzorku x_k v trénovacej množine T a $freq(C_i, T)$ je počet vzoriek triedy C_i v T.

V prípade, že vo vstupných parametroch sú i čísla, ktoré sa nedajú kategorizovať (ako ordinálne čísla, celé čísla, reálne čísla a pod., z angl. *continuous-valued*), tak je vytvorený predpoklad pravdepodobnostného rozloženia. U klasickej naivnej formy bayesovského klasifikátora je predpokladom rozloženia Gaussová distribúcia s priemerom μ a štandardnou odchýlkou σ, ktorá je definovaná ako

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} exp - \frac{(x-\mu)^2}{2\sigma^2}, \tag{3.41}$$

potom je pravdepodobnosť $P(x_k|C_i)$:

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}). \tag{3.42}$$

V rámci výpočtov je ešte potrebné vypočítať priemer μ a štandardnú odchýlku σ hodnôt atribútu A_k pre trénovaciu množinu triedy C_i.

Pri samotnej klasifikácii je pre danú klasifikovanú vzorku X vyhodnotená pravdepodobnosť $P(X|C_i)$, náležitosti X do triedy C_i. Klasifikátor rozhodne náležitosť do danej triedy C_i v tom prípade, keď pravdepodobnosť $P(X|C_i)P(C_i)$ je maximálna.

Moore, Zuev [90] a Auld [11] vo svojej práci použili tzv. *kernel estimation* v naivnej forme bayesovského klasifikátora, ktorý sa líši od naivnej formy bayesovského klasifikátora,

kde je použitá gaussova distribúcia nad spojitými prvkami. Použitý *kernel estimation* poskytuje odhad reálnej hustoty pomocou

$$P(x|C_i) = \frac{1}{n_{c_1} h} \sum_{y_j : C(y_j) = C_j} K\left(\frac{t - y_j}{h}\right), \tag{3.43}$$

kde $K(t)$ je nezáporná funkcia, ktorá spĺňa podmienku

$$\int_{-\infty}^{\infty} K(t)dt = 1.$$

a v prípade tejto práce a prác Moore, Zuev a Auld je použitá gaussova distribúcia. Použitá naivná forma bayesovského klasifikátora v rámci detekčných metód má definované dve klasifikačné triedy:

- C_V – trieda validnej komunikácie – v prípade, že analyzovaná komunikácia nie je útok, analyzovaná signatúra by mala byť klasifikovaná do tejto kategórie;
- C_U – trieda útoku – analyzovaná komunikácia, ktorá nesie prvky útoku je klasifikovaná do tejto triedy.

Vstupom do bayesovských klasifikátorov je signatúra spojenia (výstupy metrík z definícií 3.3.2 až 3.3.9) a výstupom je zaradenie (klasifikácia) do dvoch uvedených tried, teda pravdepodobnosť, že dané spojenie je validné alebo sa jedná o útok.

Korekcia nulových pravdepodobností. V prípade, že existuje trieda C_i, a X obsahuje atribút x_k, ktorý nie je obsiahnutý v žiadnej vzorke z triedy C_i v trénovacej množine T, potom pravdepodobnosť $P(x_k|C_i) = 0$. Potom i celková pravdepodobnosť $P(X|C_i)$ je rovná 0, i v prípade, že pravdepodobnosti ostatných vzoriek $P(x_k|C_i)$ pre X sú nenulové. V prípade, že nastane táto situácia, je použitá tzv. *Laplasova korekcia*, ktorá ku každému parametru z k parametrov danej vzorky pridá jeden výskyt a celkový počet $|T|$ výskytov je navýšený o k. V prípade dostatočne veľkej trénovacej množine je táto zmena nepatrná a môže byť zanedbaná.

Support Vector Machines

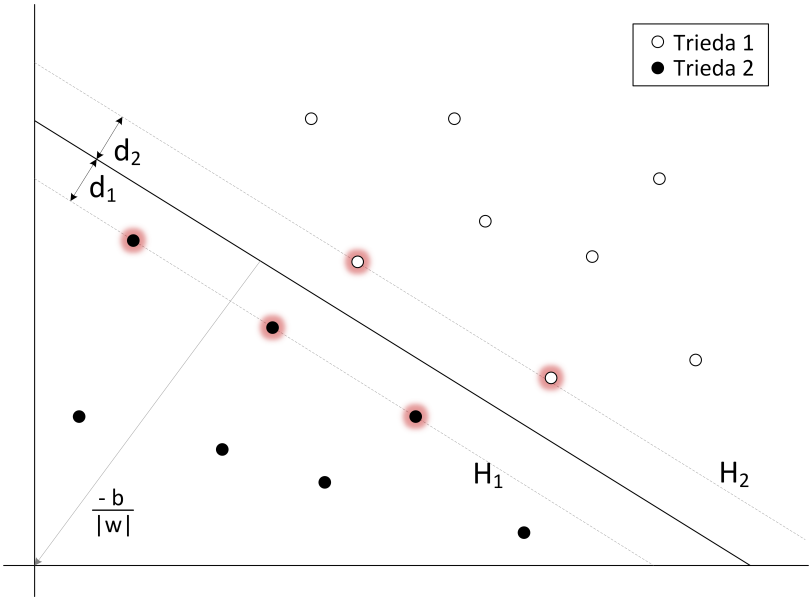
Support Vector Machines (SVM) [57] [28], tak ako ich popísal vo svojej práci Vapnik v roku 1995 [36] je metóda určená k binárnej klasifikácii prvkov (klasifikácii do dvoch tried). Majme množinu L vzoriek, kde každý vstup x_i má D atribútov (dimenzií) a je zaradený do jednej z tried $y_i = -1$ alebo $+1$. Trénovacie dáta majú teda nasledujúcu podobu:

$$\{x_i, y_i\}, \quad i = 1 \ldots L, \quad y_i \in \{-1, +1\}, \quad x \in \Re^D \tag{3.44}$$

Pri predpoklade, že dáta x_i sú lineárne oddeliteľné, môžeme x_1 a x_2 oddeliť čiarou (v prípade $D = 2$), prípadne prvky $x_1, x_2, \ldots x_D$ môžeme oddeliť hyperplochou (nadrovinou) (pri $D > 2$). Táto hyperplocha môže byť popísaná obecnou rovnicou

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \tag{3.45}$$

kde w je normála hyperplochy a $\frac{b}{||w||}$ je pravouhlá vzdialenosť od počiatku súradnicovej sústavy.



Obr. 3.3: Zobrazenie hyperplochy oddeľujúcej dve lineárne oddeliteľné triedy.

Cieľom SVM je orientácia týchto hyperplôch takým spôsobom, aby bola čo najďalej od najbližších prvkov tried, ktoré oddeľuje. Na obrázku 3.3 je zobrazená hyperplocha oddeľujúca dve lineárne oddeliteľné triedy. Podfarbené prvky tried, tzv. *Support Vectors* sú tie prvky, ktoré ležia na plochách H_1 a H_2. Definujeme d_1 ako vzdialenosť H_1 od hyperplochy a d_2 vzdialenosť H_2 od hyperplochy. Hyperplocha je ekvidistantná k plochám H_1 a H_2, čo znamená, že $d_1 = d_2$ a táto vzdialenosť je nazývaná okraj (z angl. SVM *margin*). Cieľom SVM algoritmu je nájsť takú priamku/hyperplochu, ktorá najlepšie oddeľuje dané prvky, tzn. jej vzdialenosť od najbližšieho bodu z každej triedy je najväčší, teda maximalizovať okraj hyperplochy.

Tento problém je možné formulovať ako optimalizačný problém minimalizácie vzdialenosti objektu od hyperplochy:

$$f(\mathbf{w}; b) = \frac{1}{2}||\mathbf{w}||^2 \rightarrow min. \tag{3.46}$$

Pri zachovaní klasifikácie oddeľovaných prvkov tried je nutné zachovať podmienku:

$$\forall i: \; y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0. \tag{3.47}$$

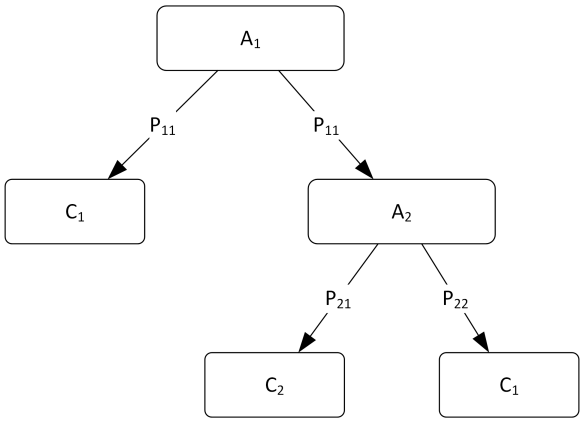
Tým je formulovaný optimalizačný problém s n obmedzujúcimi lineárnymi nerovnicami.

Od roku 1992 [23] sa rozšírila klasifikácia o tzv. *kernel* funkcie, ktoré mapujú prvky do priestoru vyššej dimenzie, kde môžu byť lepšie oddelené lineárnou funkciou, čím je možné riešiť i nelineárne problémy. Ďalší popis SVM je nad rámec tejto práce a s SVM bude ďalej pracované ako s binárnou klasifikáciou prvkov (binárna klasifikácia je postačujúca pre účely tejto práce).

Rozhodovací strom

Rozhodovací strom [131] tvorí sada hierarchicky usporiadaných rozhodovacích pravidiel, ktoré klasifikujú vstupné parametre/premenné do $n \geq 2$ tried $C_1 \ldots C_n$ a platí, že:

- každý uzol stromu (nie list) je označený niektorým z atribútov $A_1 \ldots A_m$, $m \geq 1$,
- každá hrana je označená predikátom aplikovateľným na atribút asociovaným s rodičom,
- každý list je označený niektorou triedou C_i.



Obr. 3.4: Obecný rozhodovací strom.

Vytvorenie rozhodovacieho stromu je daný algoritmom ID3 [34], kde pre celú množinu vstupných dát je vybraný atribút, ktorý má najvyšší informačný zisk (výber je daný najnižšou mierou entropie). Podľa hodnôt tohoto atribútu je koreňový uzol rozdelený na listy. Ak všetky dáta v danej podmnožine koreňového atribútu nepatria do jednej triedy, postup sa opakuje pre všetky vytvorené listy a tie sa stávajú koreňom daného podstromu (ilustrácia obecného rozhodovacieho stromu je zobrazená na obrázku 3.4).

Informačný zisk atribútu je štatistická charakteristika daného uzlu a je daná ako:

$$Z(S, A) = E(S) - \sum_{h(A)} \frac{|S_v|}{|S|} E(S_v), \tag{3.48}$$

kde $h(A)$ je množina všetkých hodnôt atribútu A, S_v je taká podmnožina množiny S, pre ktorú atribút A nadobúda hodnotu v a E je entropia (miera homogenity vstupných dát), ktorá je daná vzťahom:

$$E(S) = -\sum_{i=1}^{c} p_i log_2 p_i, \tag{3.49}$$

kde c je počet klasifikačných tried prvkov množiny S a p_i je podiel počtu prvkov i-tej triedy na celkovej kardinalite S, pričom pri výpočtoch definujeme $0.log_2 = 0$.

V prípade klasifikačného problému detekcie útokov zo sieťového toku postačia 2 klasifikačné triedy: trieda validnej komunikácie C_V a trieda útoku C_U. Pri rozhodovacom strome ako klasifikačným algoritmom je ale potrebné počítať s nestabilitou – malá zmena v dátach

môže vyvolať veľkú zmenu v rozhodovacích pravidlách, čo môže viesť k zmenám výslednej klasifikácie a s problémom pretrénovania (z angl. *overfitting*), ktorý je možno zjednodušiť na vytvorenie zložitého stromu, ktorý má výborné výsledky dosiahnuté na trénovacej vzorke, ale v prípade klasifikácie nových prvkov je úspešnosť algoritmu degradovaná.

3.7 Zhrnutie

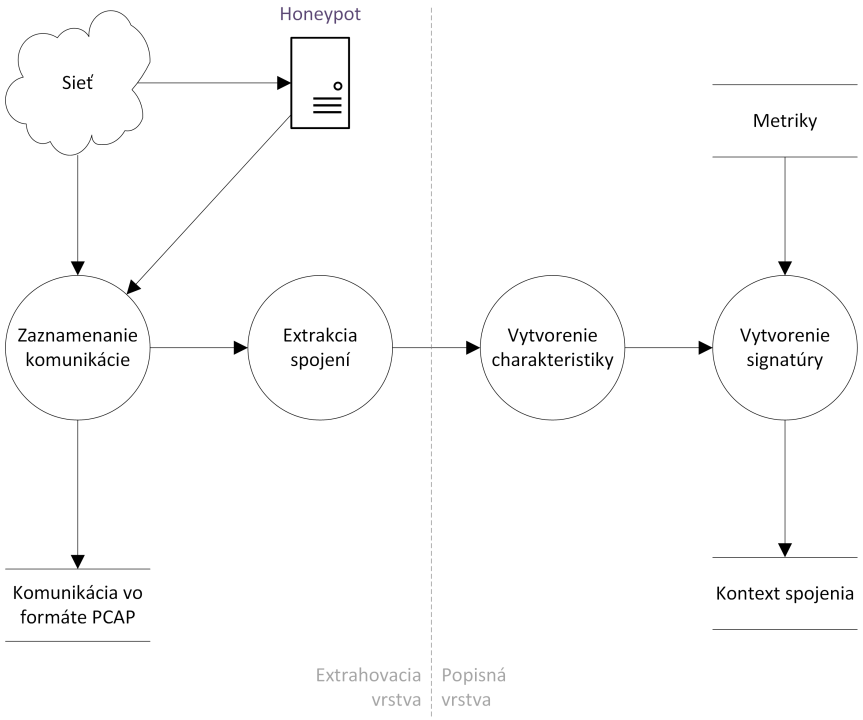
V rámci kapitoly 3 bol predstavený minimalistický sieťový paketový protokol nad ktorým boli definované metriky a celková signatúra spojenia rozšírená na protokoly architektúry TCP/IPv4. Rozborom metrík nad protokolmi TCP a IPv4 bola ukázaná reálna podoba signatúry spojenia. K tejto signatúre bolo nutné definovať kontext spojenia, ktorý je nevyhnutný (ako bude ukázané v nasledujúcej kapitole) pre detekciu pokročilejších útokov. V kapitole zaoberajúcej sa metódami analýzy spojení boli predstavené základné algoritmy extrakcie rysov (tzv. *features*) a boli popísané základné metódy použité na analýzu signatúr spojení pri detekcii sieťových útokov, ktoré budú používané v nasledujúcich kapitolách. Veľmi zjednodušene bol popísaný algoritmus *Support Vector Machine*, metóda rozhodovacieho stromu (*Decision Tree*) a *bayesovský klasifikátor*, ktoré je možné použiť na klasifikáciu vstupných dát reprezentovaných definovanou signatúrou a kontextom spojenia do tried validnej komunikácie C_V a triedy útokov C_U. Popísanými algoritmami nie je obmedzený výber klasifikačných algoritmov pre detekčný systém, je možné použiť ďalšie metódy a techniky.

V nasledujúcej kapitole bude popísaná celková architektúra detekčného systému, ktorý s využitím definovanej signatúry sieťových spojení a kontextu spojenia zdrojového a cieľového systému, vytvára základ pre detekčný sieťový systém podobný IDS systémom z kapitoly 2.

Kapitola 4

Architektúra detekčného systému

Architektúra celého systému pre detekciu útokov zo sieťového toku pozostáva z troch logických vrstiev. Prvá vrstva je **extrahovacia**, v ktorej zo sieťového toku, ktorý je rozdelený na spojenia je následne pre každé spojenie vytvorená charakteristika popisujúca dané spojenie. V druhej vrstve, **popisnej**, sú z charakteristík extrahované hodnoty metrík a tie pre dané spojenie tvoria signatúru. Tento proces je v oboch vrstvách identický, ako pre získanie signatúry spojenia z útoku na honeypot systém, tak i v prípade klasifikácie sieťového toku. V prípade signatúry získanej z honeypot systémov je táto signatúra využitá pre učiaci proces klasifikačných algoritmov. V popisnej vrstve je pre danú signatúru vytvorený kontext spojenia. Samotná klasifikácia spolu s analýzou a rozhodovacím procesom tvorí vrstvu **klasifikačnú**. Celý tento proces je zobrazený na obrázkoch 4.1 a 4.4.

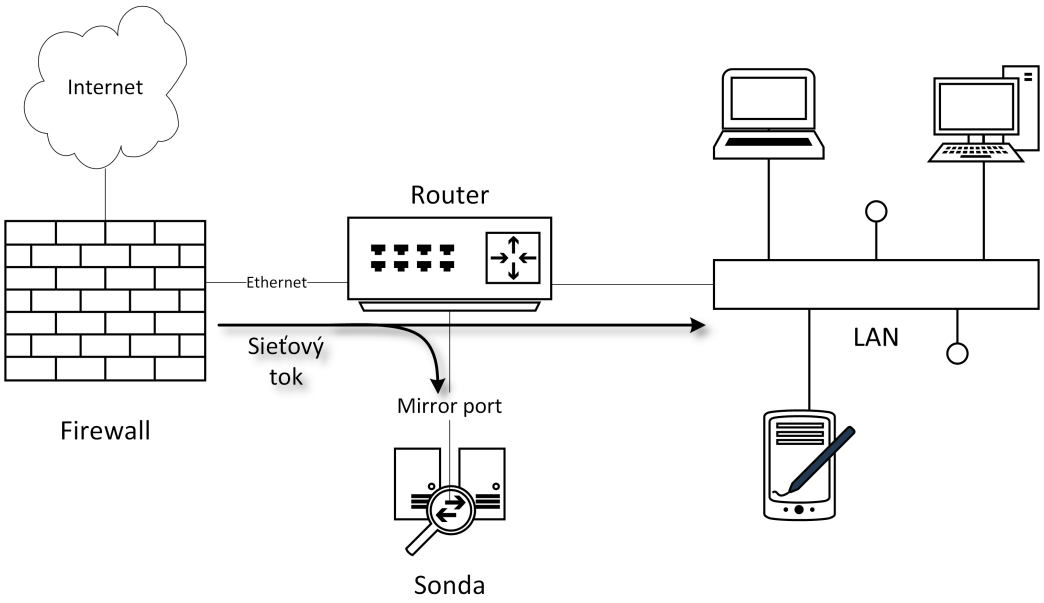


Obr. 4.1: Architektúra systému pre detekciu útokov – proces od zaznamenávania komunikácie po vytvorenie signatúry a kontextu spojenia.

4.1 Extrahovacia vrstva

Vstupom detekčného systému i extrahovacej vrstvy je záznam sieťového toku, ktorý je predmetom analýzy (budeme hovoriť o **analyzovanej komunikácii**). Tento záznam je možné získavať zo sieťového zariadenia pomocou zrkadliaceho portu[1] (pozri obrázok 4.2). Tok je ďalej možné získať i pomocou vloženia sieťového prvku (tzv. *TAP zariadenie*) medzi dve sieťové zariadenia, kde následne TAP umožní kopírovať všetok sieťový tok prenášaný medzi týmito zariadeniami a v sieti je neviditeľný (správa sa ako kábel). V neposlednom rade je možné zachytávať sieťovú komunikáciu i na koncovom zariadení.

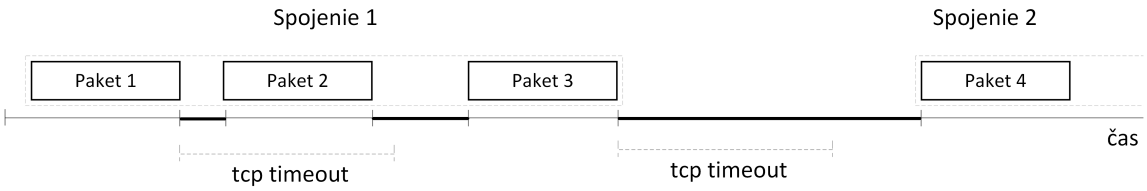
V rámci extrahovacej vrstvy je sieťový tok zaznamenávaný pomocou PCAP [69] a pre účely dodatočnej analýzy môže byť ukladaný (ide o kompletný záznam sieťového toku). Všetok zachytený tok (prúd paketov) je následne rozdelený do jednotlivých spojení, ktoré sú identifikované na základe definície 3.2.1 a rozšírenie adresy pre účely TCP/IP architektúry podľa 3.23.



Obr. 4.2: Príklad odchytávania komunikácie na routeri alebo core-switchi pri vstupe do internej siete z Internetu.

Extrakcia spojenia je daná vzťahom 3.2.1, avšak v TCP/IP neexistuje poradové číslo paketu, ale jeho poradie je možné určiť na základe sekvenčného čísla, ktoré sa používa pri zostavení celej komunikácie. Vzhľadom na to, že môžu existovať spojenia, ktoré trvajú dlhú dobu, je potrebné určiť časové okno, v rámci ktorého budú prichádzajúce pakety určené definíciou patriť do daného spojenia. Po vypršaní tohoto časového okna bude spojenie ukončené a analyzované (pozri obrázok 4.3). Pre toto časové okno nie je stanovená presná hodnota (RFC 793 [105], str. 45 definuje 5 minútový interval v rámci ktorého keď nedôjde k prijatiu ďalšieho paketu, bude spojenie zatvorené) a môže byť rôzna pre každé spojenie. V návrhu detekčného systému je potrebné s týmto počítať a stanoviť hranicu tak, aby nedochádzalo k rozdeleniu spojení, ale zároveň, aby sa nehromadili spojenia čakajúce na vypršanie časového okna z dôvodu prípadného zahltenia prostriedkov systému.

[1]Tok, ktorý prichádza do sieťového prvku je 1:1 kopírovaný na konfigurovaný port zariadenia a tým zrkadlí všetok tok do jedného výstupného portu (tzv. *mirror-port*)



Obr. 4.3: Vytvorenie spojenia č. 1 a č. 2 na základe definovaného časového okna. Medzi paketom č. 3 a paketom č. 4 dochádza k vypršaniu definovaného času a pre paket č. 4 je vytvorené nové spojenie.

V prípade honeypot systému nasadeného do monitorovanej siete je prichádzajúca komunikácia nahrávaná priamo na sieťovom zásobníku operačného systému honeypotu. V prípade, že dôjde k útoku na služby honeypot systému, sieťová komunikácia (budeme označovať ako **záznam útoku**) je zaznamenaná a odoslaná k extrakcii spojenia. Ďalšie procesy so záznamom útoku sú identické s analyzovanou komunikáciou.

4.2 Popisná vrstva

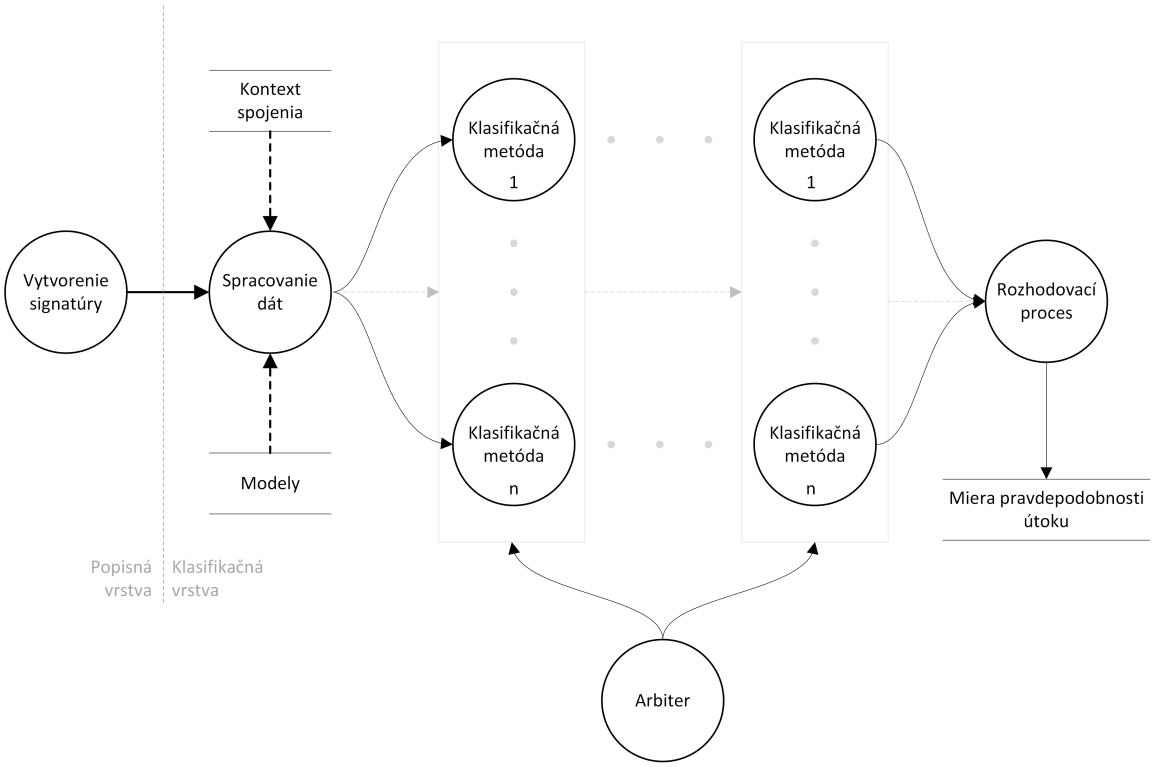
Pre každé spojenie je vytvorená charakteristika spojenia podľa 3.2.3 až 3.2.2, z tejto charakteristiky sú extrahované výstupy metrík podľa definícií 3.3.2 až 3.3.9 a tie pre dané spojenie tvoria sadu, ktorá je nazvaná *signatúra spojenia*). Táto signatúra je základným popisom analyzovaného spojenia, prípadne zaznamenaného útoku a môže byť predmetom rozšírenia systému o nové metriky. Nad rámec signatúry je možné z charakteristiky spojenia vybrať ďalšie informácie, napr.:

- hodnota, či je dané spojenie ukončené;
- identifikácia typu služby (napr. port, NBAR);
- pravdepodobnosť, že ide o šifrovanú komunikáciu;
- výstupy rôznych logických pravidiel (napr. na počet komunikácií apod.);
- ďalšie štatistické a aproximačné funkcie, atď.

Rozšírenie metrík o ďalšie informácie je predmetom budúceho výskumu. Rôzne rozšírenia môžu mať rôzne časové a výkonnostné požiadavky, niektoré metriky je možné presunúť do hardware a zrýchliť tak proces spracovania prichádzajúceho toku. Všetky uvedené vlastnosti je potrebné zohľadniť pri návrhu rozšírení systému, tak ako pri metódach analýzy v nasledujúcej kapitole.

4.3 Klasifikačná vrstva

Vstupom do klasifikačnej vrstvy je vytvorená signatúra spojenia, ktorá spolu s globálnym i lokálnym kontextom spojenia (pozri 3.5.3 a 3.5.2) je klasifikovaná voči už vytvoreným modelom analyzovaných spojení a záznamov útokov. Samotná klasifikácia na validnú a nevalidnú komunikáciu (útok), obsahuje klasifikátory, ktoré sú rozdelené do vrstiev (pozri obrázok 4.4). Jednotlivé vrstvy klasifikátorov (šedé ohraničenie) sú riadené tzv. *arbitrom*, ktorý rozhoduje o zapojení ďalšej vrstvy klasifikátora na základe definovaných pravidiel.



Obr. 4.4: Architektúra systému pre detekciu útokov – proces od vytvorenia signatúry a kontextu spojenia po výstup z detekcie.

V prípade honeypot systému je vytvorená signatúra záznamu útoku použitá ako vstup do klasifikátorov a honeypot vystupuje ako učiteľ s expertnou znalosťou. V prípade, že honeypot deteguje pretečenie zásobníka, vytvorená signatúra sieťového toku sa použije pre učiaci proces a jednotlivé modely klasifikátorov sa upravia. Pre urýchlenie procesu je možné vytvorenú signatúru porovnať s už existujúcou bázou záznamov útokov, aby nedochádzalo k učeniu a úpravám modelov vždy, keď dôjde k útoku na známu zraniteľnosť v honeypot systéme. Signatúry zachytených útokov, ktoré nie sú známe môžu byť označené a pripravené na manuálnu analýzu pre detekciu prípadných zero-day zraniteľností.

4.3.1 Klasifikačný model

Klasifikáciou v tejto práci je označovaný rozhodovací proces priradenia analyzovaného spojenia triede validných spojení alebo do triedy útokov (prípadne nevalidných alebo anomálnych spojení). Klasifikácia môže byť rozdelená do dvoch typov. V prípade, že je v rámci rozhodovacieho procesu analyzovaná komunikácia vyhľadávaná v rámci záznamov útokov (ohodnocuje sa podobnosť s útokmi) určujú klasifikátory pravdepodobnosť, že ide o útok. V prípade, že je posudzovaná miera odlišnosti analyzovaného spojenia od modelu validnej komunikácie, ide o vyhľadávanie anomálií. Vstupom do klasifikácie pre konkrétnu klasifikačnú metódu sú nasledujúce údaje:

1. podmnožina vytvorenej signatúry (pozri kapitolu 3.4.4), ktorá je tvorená konečnou množinou celých čísiel;

2. lokálny a globálny kontext analyzovaného spojenia (pozri kapitolu 3.5);

3. doplňujúce údaje – v závislosti na konkrétnej klasifikačnej metóde môžu byť dodané ďalšie voliteľné informácie (napr. existencia záznamu o analyzovanom zariadení, reputácia zdrojového a/alebo cieľového systému, validné ukončenie spojenia apod.).

Výstupom klasifikačného procesu je pravdepodobnostné ohodnotenie príslušnosti daného spojenia do kategórie útokov. Tento výstup je daný kombináciou ohodnotení (pravdepodobností) jednotlivých klasifikátorov. Ich zapojenie do procesu klasifikácie riadi arbiter.

4.3.2 Kontext spojenia

Kontext spojenia je dôležitým vstupom detekčných metód pre odhaľovanie špecifických sieťových útokov. Príkladom môže byť útok DoS – (z angl. *Denial of Service*) je dočasné alebo úplné zamedzenie poskytovania služby, dát alebo informácií validným užívateľom, zväčša ide o útok na vyčerpanie prostriedkov cieľa. Špecifickým útokom je DDoS (z angl. *Distributed DoS*), ktorý je vedený na cieľový systém väčším množstvom zdrojových systémov, zväčša tisícmi, až miliónmi. Tento útok nie je sám o sebe rozpoznateľný v prípade, že zdroj pošle na cieľový systém iba niekoľko paketov, ktoré samotné nie sú detegované, ale kontext cieľového systému obsahuje veľké množstvo krátkych spojení na rovnakú službu. Kontext je vložený do procesu spracovania dát pre jednotlivé klasifikátory, ktoré môžu využiť niektoré informácie z lokálneho alebo globálneho kontextu zdrojového alebo cieľového systému pre vlastnú klasifikáciu.

4.3.3 Arbiter

Účelom arbitra je riadenie klasifikačnej časti systému a obsluha základných funkcionalít detekčnej časti. Medzi hlavné činnosti patrí:

- Určovanie váh pri klasifikačnom procese na základe ktorých sa vyhodnocuje, či ide o útok alebo validnú komunikáciu a proces riadenia incidentu pri jeho vytvorení (na základe vyhodnotenia analyzovanej komunikácie ako útok).
- Vyhodnocovanie detekčného procesu a rozhodovanie nad ďalšími klasifikačnými úlohami. V prípade, že analyzovaná komunikácia je vyhodnotená ako podozrivá, môže byť ďalej určená na ďalšiu analýzu inou klasifikačnou vrstvou.
- Priorizácia úloh – arbiter vystupuje ako plánovač nad klasifikačnými úlohami a priorizuje jednotlivé úlohy na základe definovaných pravidiel.
- Spúšťanie procesov (pre)učenia klasifikačných metód v prípade aktualizácie databáz útokov z honeypot systémov.

4.4 Detekcia pokročilých útokov

Popisovaná architektúra detekčného systému poskytuje nástroje pre implementáciu pokročilejšej logiky detekcie útokov na sieťovej úrovni. Jednoduché detekčné algoritmy a klasifikačné metódy sú efektívne pri detekcii jednoduchých (dalo by sa povedať, že priamočiarych) foriem útokov, ktoré je možné detegovať v rámci príznakov jedného spojenia. V prípade, že ide o detekciu sofistikovanejších útokov, je potrebné vytvoriť ďalšie vrstvy detekčného systému, ktoré dokážu korelovať niekoľko parametrov a vyhodnotiť danú situáciu na komplexnejšej úrovni. Tieto pokročilé útoky môžu byť detegované len abstraktnejším pohľadom

na prebiehajúce udalosti v sieti. Jedným z prístupov môže byť multi–agentný systém, prípadne vytváranie databázy asociačných pravidiel, ktoré by modelovali na základe výsledkov nižších vrstiev (popísaných v kapitole 4.3) a na základe naučených asociácií útoky, ktoré nie sú bez dostatočnej abstrakcie možné detegovať.

Príklad: *Pri sieťovom útoku buffer overflow na sieťovú službu je útočníkovi otvorený príkazový riadok s privilegovanými právami a útočník pri kompromitácii daného systému nasadí na systém malware, ktorý v pravidelných intervaloch zasiela pakety na C&C stroj útočníka, kde očakáva ďalšie inštrukcie.*

Pri hore uvedenom príklade môže byť útok detegovaný len pri samotnej kompromitácii systému, ale navrhnutý detekčný systém bez vyššej logiky nedokáže rozpoznať vytvorenie spojenia napadnutého systému na C&C server. Toto správanie je ale možné vytvoriť ďalšími detektormi, ktorí dokážu rozpoznať anomálie nie v samotnej charakteristike spojenia, ale jej samotnou existenciou – napr. analyzovaním globálneho kontextu kompromitovaného systému. Tieto pokročilé detekcie sú nad rámec tejto práce a ďalšie úvahy sú ponechané na čitateľovi.

4.5 Zhrnutie

V predchádzajúcej kapitole bol predstavený základný návrh architektúry detekčného systému, ktorý pozostáva z troch logických vrstiev. V prvej *extrahovacej* vrstve sú analyzované toky paketov rozdelené na jednotlivé spojenia a pre každé spojenie je vytvorená charakteristika podľa 3.2.3 až 3.2.8. V *popisnej* vrstve sú z charakteristík spojení extrahované signatúry pomocou definovaných metrík. V prípade, že spojenie pochádza z honeypot systému, je táto signatúra zaradená do databázy *záznamov útokov* a použitá pre učiaci proces klasifikátorov. Signatúry analyzovaných spojení ďalej vstupujú do *klasifikačnej* vrstvy, kde prechádza spracovacím procesom, kde sa k signatúre pripojí kontext spojenia a známe (naučené) modely správania útokov. Poslednou časťou architektúry detekčného systému sú klasifikátory a arbiter, ktorý riadi klasifikačný, učiaci a rozhodovací proces.

Kapitola 5

Experimenty

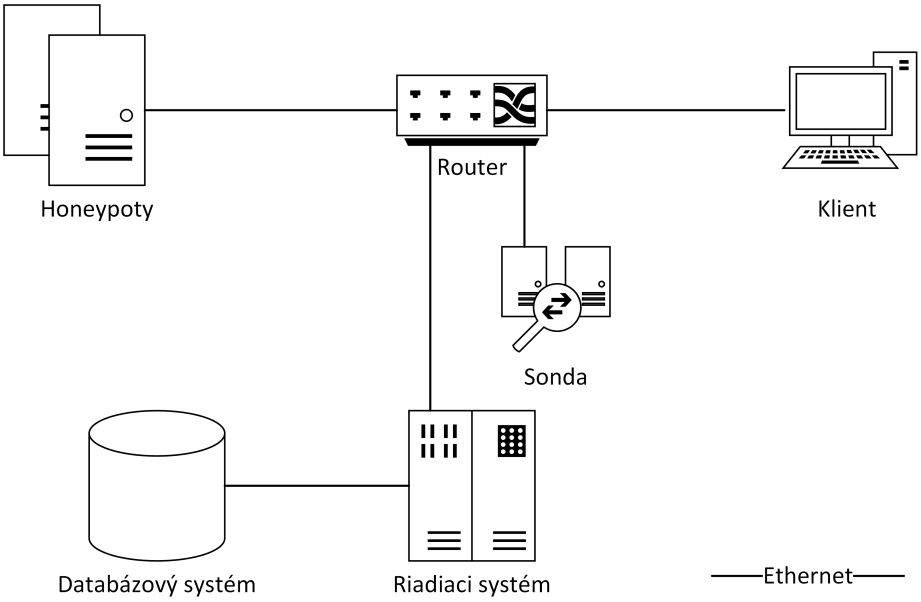
Navrhnutý systém bol v rámci overenia konceptu (z angl. *proof of concept*) čiastočne implementovaný v podobe jednoduchých programov pre spracovanie vstupných dát definovanými metrikami do dátovej sady charakterizujúcej namerané dáta. Táto sada hodnôt následne slúžila pre experimenty rôznymi metódami strojového učenia pre určenie kombinácie metód pre najlepšie vlastnosti detekcie a overenie konceptu systému simuláciami útokov a detekčné schopnosti v rámci existujúcich dátových množín. Dáta použité pri experimentoch boli získané z niekoľkých zdrojov. V rámci tejto kapitoly sú popísané zdroje dát pre experimenty, priebeh jednotlivých experimentov a dosiahnuté výsledky v nadväznosti na state-of-the-art.

5.1 Laboratórne prostredie

Pre fázu analýzy (a z časti i implementáciu) bolo vytvorené laboratórne prostredie, ktoré slúžilo na vytvorenie simulácie útokov a ich detekciu. Toto prostredie bolo vytvorené za účelom laboratórnych pokusov, dôraz bol kladený na izoláciu prostredia a procesov do takej miery, aby zaznamenané dáta pri simulovaných útokoch boli v čo najmenšej miere ovplyvnené okolím. Cieľom bolo vytvorenie dátovej sady reprezentujúcej buffer overflow útoky na sieťové služby, zaznamenanie úplného sieťového toku útoku, overenie jeho detekcie na honeypot systéme a vytvorenie charakteristiky spojenia na základe definovaných metrík.

Na obrázku 5.1 sa nachádza nákres laboratórneho prostredia, ktoré sa skladá z nasledujúcich systémov: Honeypot systémy reprezentujúce ciele simulovaných útokov založené na systéme Argos [104] obsahujúce OS Windows XP, Windows 2000 a Linuxový systém vo verzii Fedora Core 4. Riadiaci systém s databázou pre ukladanie dát a udalostí, do ktorej sa zaznamenáva sieťový tok, udalosti, obraz pamäte a procesora honeypot systémov atď. Riadiaci systém zároveň pracuje ako manažment honeypot systémov a obsahuje zdieľané úložisko pre konfiguračné práce a údržbu. Klientsky systém pre simulovanie útokov bol postavený na OS Backtrack, neskôr Kali Linux. Z tohoto systému boli vedené útoky na honeypot systémy.

V rámci laboratórneho prostredia bol vytvorený referenčný obraz (z angl. *image*) pre každý testovaný operačný systém. Pre laboratórne simulácie boli vybrané tri systémy: Windows XP, Windows 2000 a Linuxový systém distribúcie Fedora Core FC-4. Tieto systémy neboli vybrané náhodne, ale na základe požiadaviek získaných zraniteľných programov (uvedené v prílohe B), ktoré boli stiahnuté z rôznych dostupných Internetových zdrojov a to pre účely simulácie sieťových útokov. Zraniteľné verzie týchto programov boli väčšinou určené pre uvedené systémy. Každý z týchto programov obsahuje zraniteľnosť buffer overflow, ktorá



Obr. 5.1: Nákres laboratórneho prostredia.

bola vybraná pre fázu experimentov z dôvodu jej rozšírenosti a nebezpečnosti vzhľadom na riziko týchto útokov (útočník po úspešnom útoku môže získať práva zraniteľnej služby a prístup do operačného systému).

Na obrázku 5.2 je zobrazená štatistika identifikovaných buffer overflow zraniteľností. V rámci tejto štatistiky a ďalších experimentov sú brané za buffer overflow zraniteľnosti podľa rozdelenia CWE[1] nasledujúce slabiny:

- **CWE-119** – Improper Restriction of Operations within the Bounds of a Memory Buffer;
- **CWE-94** – Improper Control of Generation of Code ('Code Injection');
- **CWE-134** – Use of Externally-Controlled Format String.

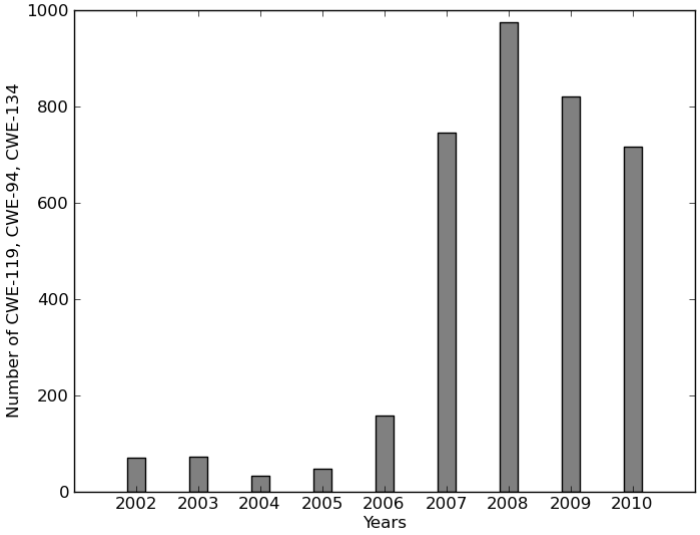
Tieto slabiny v kóde môžu viesť k buffer overflow zraniteľnosti a uvedené chyby sú odkazované v rámci CVE[2] databázy zraniteľností z ktorej bola vytvorená štatistika (štatistika založená na dátach z roku 2011) na obrázku 5.2 (vytvorené v rámci článku [46]).

Po získaní väčšieho množstva programov zraniteľné na buffer overflow (odkazujúce sa na uvedené slabiny v kóde) boli tieto programy spárované s existujúcim exploitom. Výsledná databáza zraniteľných programov je uvedená v prílohe B. Tieto programy boli uložené s referenciou na príslušný exploit do externého zdieľaného súborového systému a pre každý program bol vytvorený obraz systému z referenčného obrazu požadovaného operačného systému a nainštalovaného programu.

V rámci manažment systému bol vytvorený skript pre rotáciu systémov (vzhľadom na dostupný výkon hardware nemohli všetky systémy bežať súčasne) a tento skript zároveň

[1]CWE – Common Weakness Enumeration, ktoré popisujú slabiny v kóde ako chybné časti a programovacie technicky, ktoré spôsobili danú zraniteľnosť.

[2]CVE – Common Vulnerabilities and Exposures – zraniteľností a objavené problémy v informačnej bezpečnosti zameriavajúce sa na poskytovanie všeobecných pomenovaní pre zverejnené kyber-bezpečnostné problémy. Každá zverejnená zraniteľnosť popísaná CVE záznamom môže byť odkazovaná na CWE.



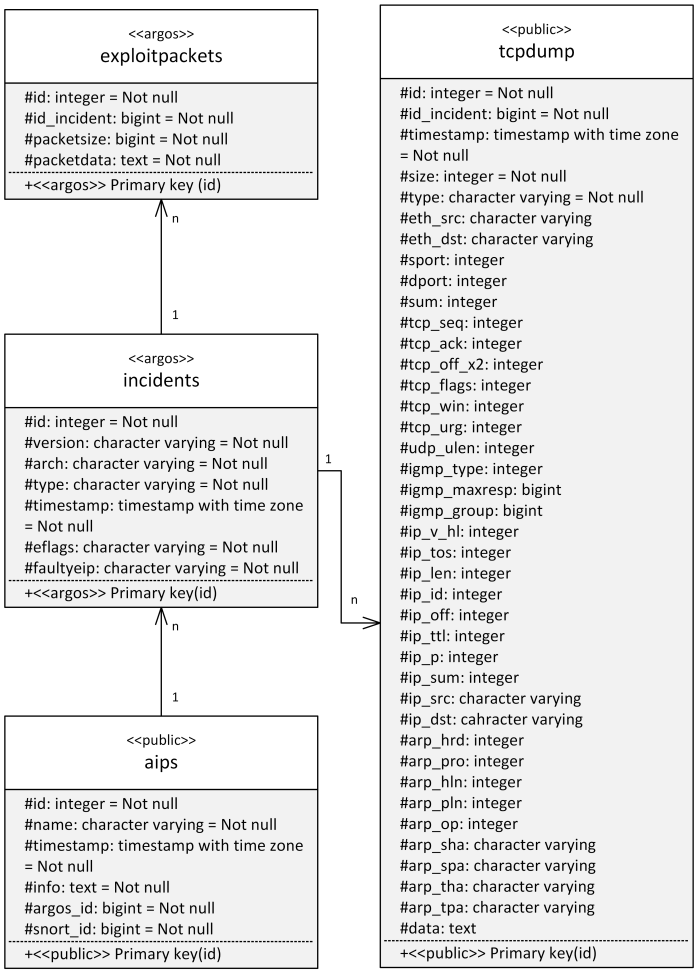
Obr. 5.2: Štatistika novo-vzniknutých zraniteľností pretečenia zásobníka (CWE-119, CWE-94, CWE-134) medzi rokmi 2002 a 2010.

spúšťal modul prispôsobeného Argos honeypot systému. Zdrojové kódy honeypot systému Argos boli upravené tak, aby získané dáta zo zachyteného útoku boli zapisované do databázy na manažment systéme (pôvodne boli vytvárané textové súbory v rámci hostiteľského systému). Systém Argos štandardne beží pod virtualizačným hypervizorom QEMU, nad ktorým bola vytvorená sada skriptov tak, aby každý simulovaný honeypot systém bežal ako služba hostiteľského operačného systému pre ľahší manažment bežiacich procesov a rotovanie cieľových systémov.

Na obr. 5.3 je zobrazená štruktúra databázy, do ktorej sú ukladané dáta z útokov na honeypot systémy. Tabuľka *exploitpackets* obsahuje pakety zachytenej komunikácie. Pri každom spojení na honeypot systém je spojenie uložené v dočasnej pamäti a v prípade detekcie útoku sú pakety tohoto spojenia vložené do tabuľky. Ku každému paketu je potom pridané unikátne číslo incidentu (ako databázový trigger pri vzniknutí nového záznamu v tabuľke *incidents*). Tabuľka *incidents* obsahuje unikátne záznamy detegovaných incidentov so základnými informáciami ako napríklad časová pečiatka (ostatné položky nie sú pre sieťovú detekciu dôležité). Obe popisované tabuľky vytvára systém Argos. Ďalšou tabuľkou je *aips*[3], kde sú uložené údaje o jednotlivých incidentoch ako číslo incidentu pre referenciu do tabuliek *exploitpackets* a *incidents*, časová pečiatka, informácie k incidentu, názov aips systému (pre prípad viacerých inštancií), ID argos honeypot systému a ID do záznamov systému snort, ktorý nebol pri experimentoch využívaný. Posledná dôležitá tabuľka je *tcpdump*, ktorá obsahuje kompletný sparsovaný záznam komunikácie spojený s incidentom. Jednotlivé položky tabuľky zodpovedajú údajom z ethernetového rámca, IP hlavičky, UDP, prípadne TCP hlavičky a dátam paketu (v určitých prípadoch ARP a IGMP atribútom).

Klientsky systém bežal na operačnom systéme Kali Linux (pôvodne Backtrack), ktorý obsahuje framework Metasploit určený na penetračné testy systémov a aplikácií (zároveň obsahoval väčšinu potrebných exploitov pre zraniteľné služby a programové vybavenie pre

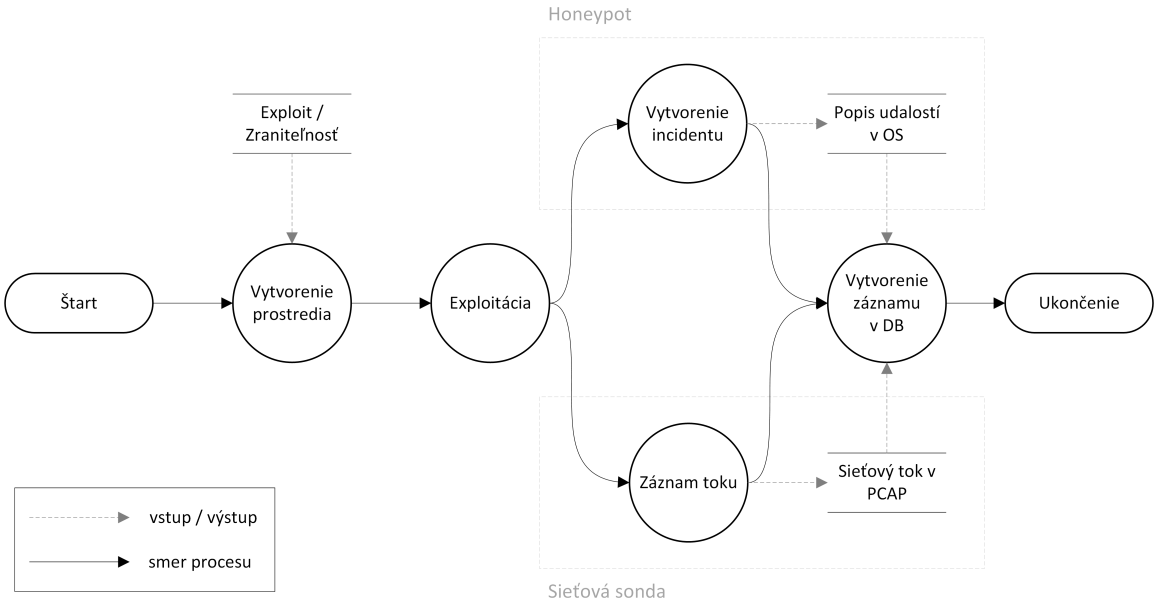
[3]AIPS je pôvodný pracovný názov pri výskumu a vývoji tohoto systému



Obr. 5.3: Schéma databázy pre spracovanie získaných udalostí, dát z cieľových honeypot systémov a zachyteného sieťového toku.

spustenie ostatných získaných exploitov). Všetok tok z/do tohoto systému bol zaznamenávaný na sieťovej úrovni a zaznamenané dáta boli ukladané v podobe PCAP [69] súborov v rámci sieťovej sondy. Pri vytvorení novej udalosti v databáze manažment systému bol automaticky spustený (pomocou triggeru v DB pri vytvorení nového záznamu v tabuľke incidentov) skript, ktorý spracoval uložený PCAP súbor so sieťovým tokom útoku pre identifikovaný tok a ten uložil do databázy pre neskoršiu analýzu. Tento tok bol v rámci simulácie jednoznačne identifikovaný a to vďaka existujúcim informáciám z cieľových systémov (IP a MAC adresa zdroja útoku, časové údaje o útoku a paket, ktorý spôsobil pretečenie zásobníka).

Celý priebeh experimentu v laboratórnom prostredí je uvedený na obrázku 5.4. Proces na strane honeypot systému a na strane sieťovej sondy po ukončenie celého procesu bol zautomatizovaný. Príprava prostredia s výberom a inštaláciou zraniteľného programu a príprave exploitu so samotnou exploitáciou bol manuálny.



Obr. 5.4: Priebeh experimentu od vytvorenia prostredia po zaznamenanie zachyteného útoku.

5.1.1 Výsledky experimentov v laboratórnom prostredí

V rámci experimentov bolo nájdených 294 zraniteľností na pretečenie zásobníka s verziami programov a existujúcim exploitom. V rámci týchto zraniteľností bolo získaných celkovo 57 programov pre Windows XP, Windows 2000 a Linux. V rámci experimentov boli vykonané útoky podľa existujúceho exploitu (zväčša použité nástroje Metasploit, nástroje systému Backtrack a exploity z databáz exploitdb a podobných) a po úspešnej exploitácii služby bol tok spojený s útokom zaznamenaný do databázy pre ďalšiu analýzu.

Výsledky simulácií boli neskôr použité pri návrhu detekčného systému a experimenty s rôznymi sadami metrík. Spočiatku boli získané dáta použité i na experimenty s rôznymi klasifikačnými metódami a metódami strojového učenia, ale nazbierané vzorky dát neboli pre tieto úlohy dostatočne kvalitné (hlavne kvôli malému počtu úspešných útokov, nízkej diverzite útokov a homogénnosti laboratórneho prostredia). Popis experimentov, dátová sada a podrobný popis metrík sú uvedené v rámci článku *Behavioral signature generation using shadow honeypot* uverejnenom v roku 2012 na *International Conference on Computer Networks and Systems Security* organizácie World Academy of Science, Engineering and Technology v Tokiu [13].

V rámci experimentov so simulovanými útokmi v laboratórnom prostredí boli použité základné štatistické metriky, ktoré boli doplnené aproximačnými metódami (gaussové krivky, polynomiálna aproximácia a Fourierova transformácia), celkovo 169 metrík. Na overenie klasifikačných algoritmov nad experimentálnou sadou dát bol použitý program *RapidMiner* [112] a to s metódami uvedenými v tabuľke 5.1.

Výsledky experimentov sú porovnateľné s výsledkami obdobných vedeckých prác, i keď kvalita testovacích dát nebola uspokojivá. Útoky boli vedené z jedného klientskeho počítača a IP adresa sa počas pokusov nemenila, to isté je možné povedať o cieľových honeypot systémoch. Metriky založené na adresácii z tohto dôvodu boli z experimentov vylúčené. V prípade, že by boli pri klasifikácii ponechané, bol by klasifikačný proces znehodnotený

Klasifikačná metóda	SVM	Rozhodovací strom	Bayesovský klasifikátor s PCA s automatickým počtom komponentov	Bayesovský klasifikátor s diskretizáciou ordinálnych parametrov	Naivný bayesovský klasifikátor	Naivný bayesovský klasifikátor a PCA s fixným počtom komponentov
Účinnosť	41,67 %	25,00 %	16,67 %	25,00 %	8,33 %	8,33 %
Špecifickosť	96,09 %	94,97 %	94,48 %	94,97 %	96,13 %	96,73 %
Precíznosť	41,67 %	25,00 %	16,67 %	25,00 %	14,29 %	16,67 %
Presnosť	89,85 %	87,82 %	87,82 %	87,82 %	76,14 %	75,63 %

Tabuľka 5.1: Prehľad výsledkov experimentov s klasifikačnými metódami nad laboratórnymi dátami.

práve jednoznačnou identifikáciou útočníka a obeti (validná komunikácia nebola vedená na honeypot systémy). Počet nahraných útokov (celkovo iba v rámci 57 získaných zraniteľných programov) nebol postačujúci pre kvalitnú dátovú sadu na klasifikačné a testovacie experimenty (výberom programov, ktoré boli zraniteľné a zároveň poskytovali sieťovú službu, redukciou útokov, ktoré boli rovnaké alebo obdobné a rozdelením sady útokov na niekoľko množín pre učenie, validáciu a testovanie sa konečný počet redukoval na 11 unikátnych útokov na množinu). Pri vytváraní validných spojení boli simulované validné spojenia na služby, ktoré boli pri útokoch napádané a preto boli tieto spojenia veľmi podobné záznamom útokov, avšak nezodpovedajú reálnym podmienkam. Pre uspokojivú veľkosť databázy pre vierohodné testovanie klasifikačných algoritmov by bolo nutné simulovať rádovo stovky, až tisíce útokov a k tomu prislúchajúce validné spojenia v minimálne rovnakom počte.

V tabuľke 5.1 sú uvedené výsledky jednotlivých klasifikačných algoritmov. Celková **účinnosť** (čiže pomer správne identifikovaných útokov voči všetkým správne identifikovaným) algoritmov je veľmi nízka z dôvodu vysokého false-positive spôsobeného veľkým počtom vzoriek validných spojení voči vzorkám záznamov útokov. Toto tvrdenie dokazuje vysoká **špecifickosť** (pomer správne identifikovaných validných spojení voči všetkým záznamom validných spojení), ktorá je daná vysokým počtom záznamov validných spojení (priemerne 16-násobok k počtom záznamov útokov). Pri porovnaní klasifikačných algoritmov je kvalitatívne najlepším ukazovateľom **presnosť** klasifikácie (pomer správne identifikovaných vzoriek voči všetkým vzorkám), kde najlepšie výsledky dosiahol, so skoro 89 % presnosťou, SVM algoritmus. Tento výsledok je pre nízky pomer záznamov útokov voči všetkým záznamom očakávaný.

Pre neuspokojivé výsledky so simulovanou dátovou sadou sa výskum uberal dvoma smermi, v prvom išlo o nájdenie vhodnej verejnej dátovej sady, ktorá by spĺňala predpo-

klady pre kvalitný vstup testovania klasifikačných algoritmov a záznam reálnej komunikácie v prostredí kampusu VUT.

5.2 Prostredie VUT v Brně

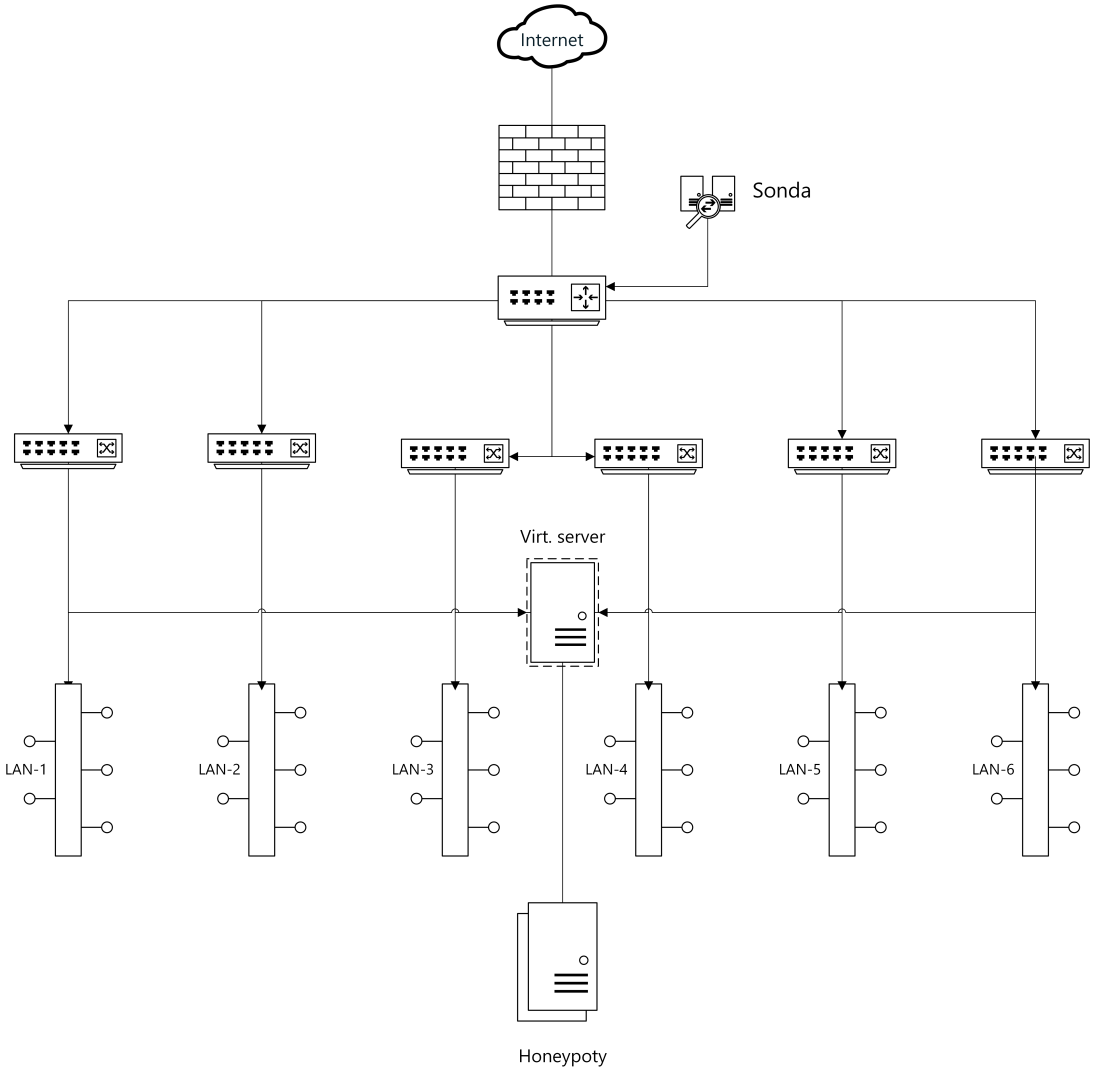
V rámci univerzitného prostredia Vysokého učení technického v Brně bola vytvorená infraštruktúra, ktorá má k dispozícii sieť s približne 20 000 užívateľmi, s priemerne 6 000 aktívnymi IP adresami ktorí generujú tok priemerne 8 TB za jeden deň, čo predstavuje približne 8 GB *NetFlow* záznamov za deň. Univerzitná sieť poskytuje dostatočne heterogénne prostredie pre zbieranie a analýzu dát. V rámci kampusu VUT boli nasadené sondy do vnútornej siete (na uzlové prvky jednotlivých VLAN) a na hraničný prvok s vonkajšou sieťou. Do všetkých VLAN boli nasadené i honeypot systémy, ktoré zbierali informácie o aktuálne prebiehajúcich útokoch (pozri obrázok 5.5). V každej zo 6 podsietí je vyvedené rozhranie jedného virtuálneho honeypotu a dáta boli zbierané na koncových honeypotoch a na sieťovej sonde.

Najväčšiu časť prenesených dát v rámci spojení s Internetom predstavoval HTTP (67 %) a SSH (2,6 %). Zároveň ale boli detegované protokoly, ktoré historicky obsahujú veľké množstvo zraniteľností, ako napríklad Netnews (2,5 %), FTP (0,8 %) atď. V rámci vnútornej siete dosahuje veľkosť prenesených dát 18 GB denne na každý sieťový smerovač, ktorý pokrýva VLAN s približne 500 uzlami. Väčšinu toku tvoria rôzne peer-to-peer siete, komunikátory, VoIP a služby na zdieľanie súborov. Jedným z najčastejších protokolov v rámci zachyteného toku predstavovali DNS záznamy (6,8 %), čo je spôsobené prítomnosťou DNS serverov v lokálnej VLAN. Ďalším najčastejším protokolom bol NetBIOS, ktorý bol zachytávaný iba v rámci vnútornej siete (NetBIOS nie je povolený na hraničných prvkoch). Ďalšie sieťové služby, ktoré sa často vyskytovali boli Alias Service (port 1187), Cache (port 1972), AMS služba (porty 1037 a 1038) a veľa ďalších.

Honeypot systémy generovali priemerne skoro 3 incidenty denne, zväčša išlo o infikované Windows XP systémy univerzitného kampusu, ktoré obsahovali malware *Conficker*. Tento malware pre svoje šírenie sa konštantne snažil napádať Microsoft Directory Services (port 445) ostatných systémov v sieti. Honeypoty po dobu približne jedného roka nedetegovali žiadne ďalšie útoky a pre svoju časovo náročnú údržbu boli vypnuté.

5.3 Výskumné databázy

Jednou z dôležitých častí pri posúdení metód klasifikácie sú dáta, ktoré sú jednoznačné (existuje znalosť o každom spojení, či ide o útok alebo validnú komunikáciu) a zároveň autentické (dáta nevykazujú skreslenie na základe ich pôvodu). V rámci výskumu *1999 KDD Cup* [64], ktorý sa uskutočnil v roku 1999 bola vytvorená databáza sieťových tokov, ktoré obsahujú útoky na sieťové služby. Táto databáza bola určená pre výskumné účely sieťových detekčných algoritmov a je široko najpoužívanejšia databáza vo výskumných projektoch [124]. I napriek tomu, že veľa expertov na systémy pre detekciu prienikov do siete uvádza, že väčšina aktuálnych útokov sú variantmi už známych útokov a signatúry pôvodných útokov sú postačujúce pre ich detekciu, práve projekt *1999 KDD Cup*, tzv. *Classifier Learning Contest* vyvrátil toto tvrdenie [48]. Cieľom ďalšieho výskumu bolo nadviazať na tento projekt, vytvoriť a poskytnúť databázu obdobnú databáze HoAH [94], ale s rozdielnou úrovňou detailu. Zmyslom vytvorenia takejto databázy bolo vytvoriť množinu dát, ktorá bude obsahovať i nové, zložitejšie útoky s maximálnou možnou úrovňou detailu (v ideálnom prípade



Obr. 5.5: Schéma univerzitnej siete VUT so zapojením honeypot systémov a sieťovej sondy.

bez straty informácie).

V roku 2009 bola vytvorená sada CDX 2009 [115], ktorá obsahovala záznamy útokov na vytvorené virtualizované prostredie (konkrétne na 4 typy služieb, pozri tabuľku 5.2)[4]. Záznamy komunikácie obsahovali externé IP adresy systémov (cieľov útokov) a záznamy snort IDS systému, ktorý detegoval útoky na tieto systémy, ale odkazoval na systémy ich internými IP adresami. Vzhľadom na fakt, že experimenty boli simulované voči buffer overflow útokom, učenie a následné testovanie navrhnutého systému a detekčných metód bolo smerované na tento typ útoku, ktorý bol vedený len voči službám *Postfix Email* a *Apache Web Server*. Záznamy zo snort IDS systému museli byť v prvej fáze mapované na záznamy sieťovej komunikácie, ktoré boli poskytnuté v separátnych súboroch. Z jednotlivých záznamov boli vybrané tie, ktoré odkazovali na buffer overflow, z ktorých boli použité polia IP adresy, porty, čas výskytu, TCP sekvenčné číslo (*Seq*) a tzv. acknowledgment číslo (*ack*).

[4]Databáza CDX 2009 bola vybraná pre experimenty z dôvodu najvernejšej simulácie útokov a reálneho prostredia [115] oproti KDD Cup a DARPA dátovým sadám.

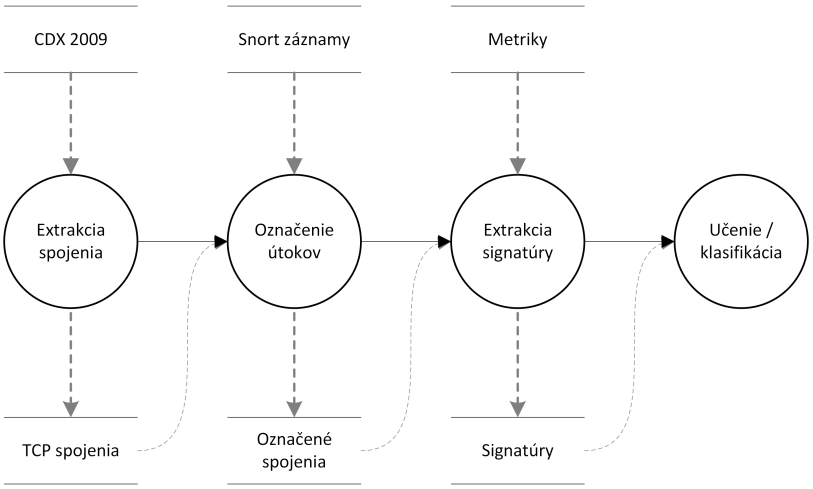
Služba	OS	interná IP	externá IP
Postfix Email	FreeBSD	7.204.241.161	10.1.60.25
Apache Web Server	Fedora 10	154.241.88.201	10.1.60.187
OpenFire Chat	FreeBSD	180.242.137.181	10.1.60.73
BIND DNS	FreeBSD	65.190.233.37	10.1.60.5

Tabuľka 5.2: Zoznam zraniteľných serverov v dátovej sade CDX 2009.

Príklad záznamu z IDS systému snort:

```
[**] [124:4:1] (smtp) Attempted specific command
buffer overflow: HELO, 2320 chars [**]
[Priority: 3]
11/09-14:22:25.794792
10.2.195.251:2792 -> 7.204.241.161:25
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:2360
***AP*** Seq: 0x68750738  Ack: 0x24941B59
Win: 0xFDC0  TcpLen: 20.
```

Postup mapovania žiaľ priniesol len 44 záznamov útokov typu buffer overflow z celkových uvádzaných 65. Tieto identifikované útoky boli následne použité na proces učenia.



Obr. 5.6: Proces extrakcie spojení z CDX 2009 dátovej sady.

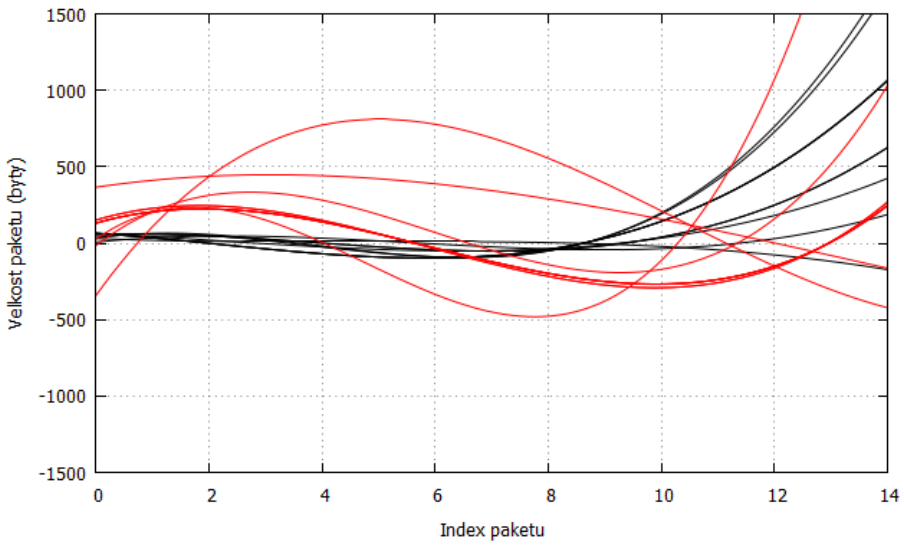
V rámci tohoto procesu bolo nutné mapovať interné adresy a externé adresy, pretože záznamy sieťového toku a záznamy zo snort systému, ako bolo popísané vyššie, používali inú adresáciu. Pre zníženie celkového počtu (4 milióny) paketov boli brané do úvahy iba 2 súbory[5], ktoré obsahujú 44 úspešne mapovaných buffer overflow útokov (celkovo 1 538 182 paketov). Tento počet bol ešte dodatočne (z dôvodu zvyšujúcej sa hustoty paketov v čase útokov) zredukovaný tak, aby obsahoval rovnomerne rozložené spojenia na konečný počet 204 953 paketov. Na obrázku 5.6 je zobrazený proces extrakcie spojení z CDX 2009 dátovej sady, označenie spojení na útoky a validnú komunikáciu podľa snort záznamov, extrakciu signatúr podľa definovaných metrík a následne proces učenia a klasifikácie, ktorý prebiehal v programe RapidMiner. Tento proces bol vykonaný nad metrikami definovanými v rámci tejto práce a nad diskriminátormi definovanými v publikácii Moore, a kol. [89].

[5]Pre úplnosť uvádzame názvy použitých súborov pre experimenty: *2009-04-21-07-47-35.dmp, 2009-04-21-07-47-35.dmp2*

Klasifikácia pomocou definovaných metrík bola vykonaná bayesovským klasifikátorom s 5-násobnou krížovou validáciou pre každý experiment s výberom rysov pomocou SFFS. Na záver boli porovnané výsledky oboch výskumných prác a prehľad najlepších metrík a diskriminátorov je možné vidieť na obrázku 5.14. V rámci experimentu boli dosiahnuté zaujímavé výsledky s aproximačnými funkciami. Najlepšia celkovú presnosť detekcie bola dosiahnutá pomocou metrík polynomiálnej aproximácie a to hlavne pri aproximácii prichádzajúceho smeru (od zdroja k cieľu) polynómom 3. a 5. stupňa. Dobré výsledky boli dosiahnuté i aproximáciou gaussovými krivkami a Fourierovými koeficientami. Ďalšie experimenty s klasifikačnými algoritmami vrátane naivného bayesovského klasifikátora sú uvedené v nasledujúcej kapitole.

5.4 Experimenty s detekčnými metódami

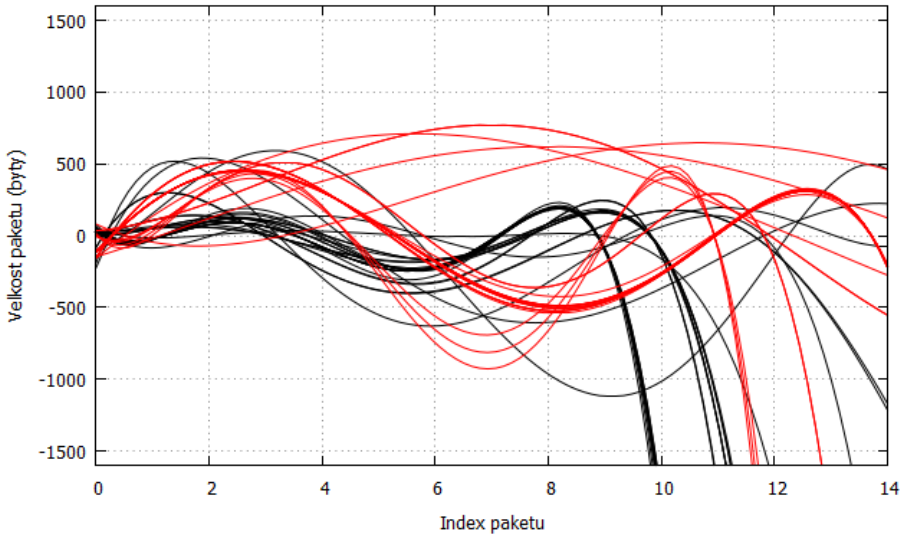
Pre overenie princípov a konceptu detekčného systému boli vytvorené experimenty s detekčnými metódami, hlavne zamerané na klasifikačné algoritmy a metódy strojového učenia. Medzi testované metódy bol zahrnutý naivný bayesovský klasifikátor s výberom rysov pomocou SFFS a metódy rozhodovacieho stromu. Algoritmus SVN bol z dôvodu výkonnostných obmedzení z experimentov vynechaný. Cieľom experimentov bolo potvrdenie hypotézy o úspešnosti klasifikácie použitím aproximačných funkcií na priebeh spojenia (behaviorálnou signatúrou).



Obr. 5.7: Polynomiálna aproximácia veľkosti paketov komunikácií na port 80 z CDX dátovej sady polynómom 3. stupňa (červenou farbou zobrazené útoky, čiernou farbou zobrazená validná komunikácia).

Vstupné dáta – ako vstupné dáta bola použitá dátová sada CDX 2009 (popis spracovania dát bol popísaný v kapitole 5.3) a to hlavne z dôvodu zlej kvality dát z laboratórnych experimentov a problematický zber dát z reálneho prostredia kampusu VUT v Brně. V rámci experimentov boli výsledky porovnávané s publikovanými diskriminátormi (pozri Moore a kol. [89]).

Ako je vidieť i na výsledkoch experimentov v nasledujúcich kapitolách, hlavné očakávania boli kladené na metriky aproximačných metód, ktoré plynú z predpokladu, že týmito metódami sa aproximuje správanie daného spojenia a je možné toto správanie klasifikovať podľa vytvorených koeficientov. Ako je vidieť na obrázkoch 5.7 a 5.8, validná komunikácia daného protokolu vykazuje stálosť jej priebehu, naopak útoky na sieťové služby cez tento protokol vykazujú značne rozdielny priebeh. Na obrázku 5.14 je vidieť účinnosť týchto metrík voči ako ostatným metrikám v definovanej signatúre spojenia tak voči spomínaným diskriminátorom.



Obr. 5.8: Polynomiálna aproximácia veľkosti paketov komunikácií na port 80 z CDX dátovej sady polynómom 5. stupňa (červenou farbou zobrazené útoky, čiernou farbou zobrazená validná komunikácia).

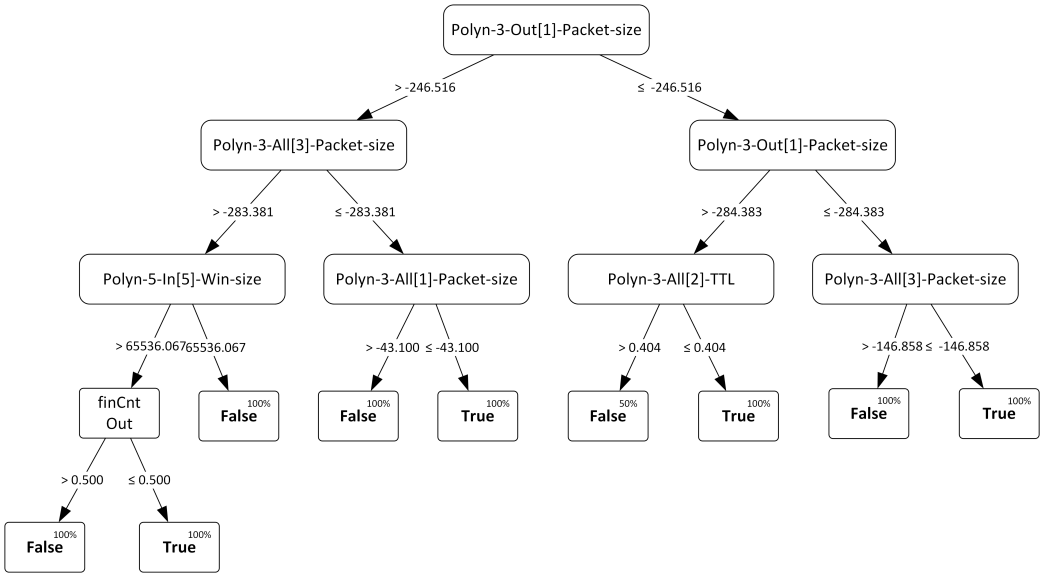
Do experimentov boli zaradené rôzne polynomiálne aproximácie, diskrétna Fourierova transformácia, ale i napríklad aproximácia gaussovými krivkami, konkrétne boli testované nasledujúce parametre uvedených funkcií:

- Aproximácia polynómom 3., 5., 8., 10. a 13. stupňa. Výpočet vyšších rádov polynómov bol časovo zdĺhavý pre ďalšie experimenty;
- 10 koeficientov diskrétnej Fourierovej transformácie;
- Aproximácia gaussovými krivkami;

5.4.1 Rozhodovací strom

Medzi experimenty s klasifikačnými metódami bol zaradený i algoritmus rozhodovacieho stromu. Tento algoritmus trpí nedostatkami, ktoré je možné spôsobiť zlým návrhom a použitými parametrami výsledného algoritmu. Rozhodovací strom, ktorý správne klasifikuje trénovacie vstupy môže byť tzv. „pretrénovaný“ (z angl. *overfit*) a vysoká úspešnosť klasifikácie môže byť znížená u nových vzoriek. Pre zlepšenie tohoto výsledného vytvoreného stromu je možné nastavovať niekoľko parametrov:

- **Pravidlo výberu** – Prvým parametrom je kritérium výberu podmienky rozhodovacieho pravidla na základe ktorého budú vyberané atribúty pre delenie (vytvorenie



Obr. 5.9: Schéma rozhodovacieho stromu s minimálnym informačným ziskom pri delení 0,1.

nových listov). Možný je výber variant na základe najnižšej entropie, použitie tzv. *Gini indexu* [47] a na základe presnosti klasifikácie celého stromu. Pre experimenty bola použitá forma najnižšej entropie.

- **Maximálna hĺbka** – Hĺbka stromu je počet vrstiev, ktoré sú algoritmom vytvárané, pre experimenty nebola stanovená pevná hranica maximálnej hĺbky stromu.
- **Prerezávanie** – Prerezávanie (z angl. *pruning* je metóda pri ktorej listy, ktoré nepridávajú dostatočnú hodnotu diskriminácie vstupov sú zo stromu v ďalšom kroku odobrané. Existuje niekoľko metód prerezávania stromu, v rámci experimentov bola použitá metóda *J-Pruning* [24].
- **Minimálny informačný zisk** – Informačný zisk sa počíta na základe entropie vstupných atribútov. Atribút s najvyšším informačným ziskom je vybraný a pri jeho delení (na listy) sa vypočítava informačný zisk každého atribútu. Pri výbere listu pred jeho delením sa rozhoduje, či delenie prebehne na základe výpočtu jeho zisku. Ak je tento zisk menší ako nastavený minimálny zisk, list sa ďalej deliť nebude. Čím vyššie je táto hodnota nastavená, tým menej listov má výsledný rozhodovací strom.

Experimenty s algoritmom rozhodovacieho stromu viedli k výsledkom, ktoré dokázali klasifikovať vstupy získané z dátovej sady CDX 2009 až s celkovou presnosťou 99,76 % (pozri tabuľku 5.3). Celkové dosiahnuté F-score u najlepšieho (podľa účinnosti klasifikácie) stromu bolo 83 %. Atribúty vybrané pri vytváraní rozhodovacieho stromu:

- *Polyn-3-Out[1]-Packet-size* – 2. koeficient polynomiálnej aproximácie variabilného atribútu veľkostí paketov odchádzajúceho spojenia polynómom 3. stupňa;
- *Polyn-3-All[3]-Packet-size* – 4. koeficient polynomiálnej aproximácie variabilného atribútu veľkostí všetkých paketov bez rozlíšenia smeru polynómom 3. stupňa;

- *Polyn-5-In[5]-Win-size* – 6. koeficient polynomiálnej aproximácie variabilného atribútu veľkosti okna v prichádzajúcich paketoch polynómom 5. stupňa;
- *Polyn-3-All[1]-Packet-size* – 2. koeficient polynomiálnej aproximácie variabilného atribútu veľkostí paketov bez rozlíšenia smeru polynómom 3. stupňa;
- *Polyn-3-All[2]-TTL* – 3. koeficient polynomiálnej aproximácie variabilného atribútu TTL bez rozlíšenia smeru polynómom 3. stupňa;
- *finCnt Out* – počet *FIN* flagov v odchádzajúcich paketoch.

	Validná kom. (false)	**Útok (true)**	**Senzitivita**
False	5723	10	99,83 %
True	4	34	89,47 %
Účinnosť	99,93 %	77,27 %	

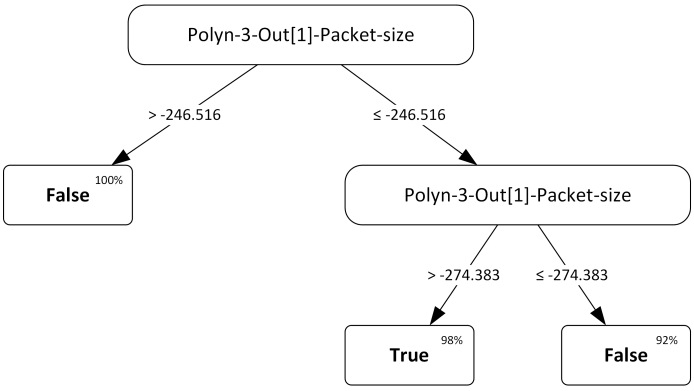
Tabuľka 5.3: Výsledky experimentov s rozhodovacím stromom s celkovou dosiahnutou presnosťou 99,76 % a celkovým F-Score 83 %.

V obrázkoch 5.9 a 5.10 sú uvedené dva výsledné rozhodovacie stromy, ktoré sa líšia v parametroch upravovaných v rámci jednotlivých experimentov a celkovou účinnosťou algoritmov v klasifikácii. U oboch stromov bolo použité prerezávanie s výberom na základe najnižšej entropie, bez obmedzenia hĺbky výsledného stromu. Strom na obrázku 5.9 bol vytvorený s minimálnym informačným ziskom medzi iteráciami 0,1 a strom na obrázku 5.10 bol vytvorený s 0-vým prírastkom informačného zisku. V prípade 2. stromu nastavený informačný zisk na 0 (nedôjde k žiadnemu deleniu) spôsobilo, že algoritmus koreňový uzol i napriek tomu rozdelil (pri zisku tohoto listu – 0) a upravil váhy rozhodovacieho pravidla. Vstupný atribút s najvyšším informačným ziskom bol 2. koeficient polynomiálnej aproximácie variabilného atribútu veľkostí paketov prichádzajúceho spojenia polynómom 3. stupňa. Celková presnosť tohoto stromu je 99,71 % s celkovým F-score 78 % (pozri tabuľku 5.4).

	Validná kom. (false)	**Útok (true)**	**Senzitivita**
False	5724	14	99,76 %
True	3	30	90,91 %
Účinnosť	99,95 %	68,18 %	

Tabuľka 5.4: Výsledky experimentov s rozhodovacím stromom s celkovou dosiahnutou presnosťou 99,71 % a celkovým F-score 78 %.

Rozhodovací strom pri zvyšovaní parametrov ako minimálny informačný zisk, či hĺbka stromu je možné naučiť na veľkú presnosť. U dát z testovacieho prostredia bolo možné naučiť rozhodovací strom na 100 % špecifickosť klasifikácie (správna klasifikácia validnej komunikácie) alebo 100 % senzitivitu (správna klasifikácia útokov), ale chybovosť takýchto stromov sa prejaví pri klasifikácii daných vzoriek i trocha odlišných v atribútoch porovnávacích pravidiel od vzoriek z pôvodnej trénovacej množiny. Tieto stromy mali hĺbku 9 úrovní



Obr. 5.10: Schéma rozhodovacieho stromu s minimálnym informačným ziskom pri delení 0.

a takéto stromy sa nazývajú pretrénované a v reálnej klasifikácii sú nepoužiteľné. Obecne platí, že výsledný rozhodovací strom by mal byť čo najjednoduchší i v prípade poklesu špecifickosti alebo senzitivity klasifikácie pre jeho lepšiu klasifikáciu odlišných vzoriek.

5.4.2 Bayesovský klasifikátor

Naivný bayesovský klasifikátor je jedným z najpoužívanejších binárnych klasifikačných algoritmov. Dôvodom je jeho rýchlosť, ktorú dosahuje jednoduchým výpočtom nad vstupnými dátami – výsledná aposteriórna pravdepodobnosť je daná súčinom a priori pravdepodobností definovaných tried a skúmaných atribútov vstupnej dátovej množiny. Pre daný atribút je vypočítaná a priori pravdepodobnosť pre obe triedy (pozri kapitolu 3.6.2).

V rámci experimentov s algoritmom naivného bayesovského klasifikátora bol (v rámci najlepších výsledkov) použitý Forward Selection (SFFS) algoritmus pre výber najlepších rysov a to s diskretizáciou do 5 košov a 5 krížových validácií. Bayesovský klasifikátor bol v režime greedy (hladného algoritmu [35]) s 10 kernelmi. Celková dosiahnutá presnosť bola 99,83 % a F-score 87,81 % (pozri tabuľku 5.5). Zaujímavé výsledky dosiahol ten istý algoritmus s jedným rozdielom a to použitím Z-transformácie pre normalizáciu prvkov. Dosiahnuté výsledky ale neboli lepšie (celková dosiahnutá presnosť 99,81 % a F-score 86,42 %).

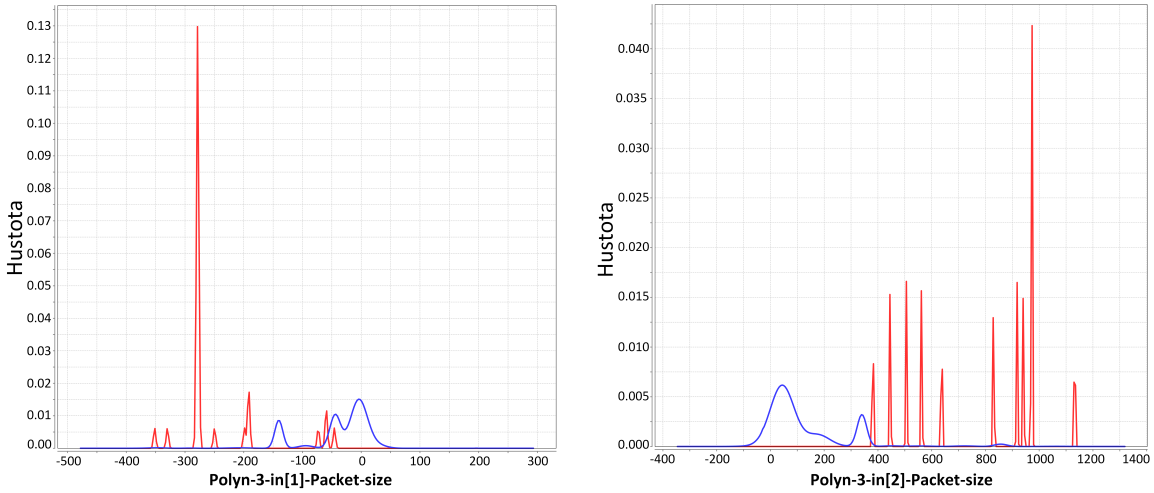
	Validná kom. (false)	**Útok (true)**	**Senzitivita**
False	5725	8	99,86 %
True	2	36	94,74 %
Účinnosť	99,97 %	81,82 %	

Tabuľka 5.5: Výsledky experimentov s bayesovským klasifikátorom s celkovou dosiahnutou presnosťou 99,83 % a celkovým F-score 87,81 %.

Vybrané najlepšie rysy uvedeného experimentu s bayesovským klasifikátorom sú uvedené nižšie. Jednotlivé rysy nie sú zoradené podľa žiadneho špecifického kľúča.

- *finCnt Out* – počet *FIN* flagov v odchádzajúcich paketoch;
- *synCnt Out* – počet *SYN* flagov v odchádzajúcich paketoch;

- *Polyn-3-All[3]-Packet-size* – 4. koeficient polynomiálnej aproximácie variabilného atribútu veľkostí všetkých paketov bez rozlíšenia smeru polynómom 3. stupňa;
- *Polyn-3-All[2]-TTL* – 3. koeficient polynomiálnej aproximácie variabilného atribútu TTL paketov bez rozlíšenia smeru polynómom 3. stupňa;
- *Polyn-3-Out[1]-Packet-size* – 2. koeficient polynomiálnej aproximácie variabilného atribútu veľkostí všetkých odchádzajúcich paketov polynómom 3. stupňa;
- *Polyn-5-Out[2]-Win-size* – 3. koeficient polynomiálnej aproximácie variabilného atribútu veľkostí okna spojenia odchádzajúcich paketov polynómom 5. stupňa;
- *Polyn-5-In[5]-Win-size* – 6. koeficient polynomiálnej aproximácie variabilného atribútu veľkostí okna spojenia prichádzajúcich paketov polynómom 5. stupňa;
- *Polyn-5-In[0]-TTL* – 1. koeficient polynomiálnej aproximácie variabilného atribútu TTL spojenia prichádzajúcich paketov polynómom 5. stupňa;
- *Polyn-3-In[1]-TTL* – 2. koeficient polynomiálnej aproximácie variabilného atribútu TTL spojenia prichádzajúcich paketov polynómom 5. stupňa.



Obr. 5.11: Distribučné rozloženie hodnôt 1. a 2. koeficientu polynomiálnej aproximácie veľkostí prichádzajúcich paketov.

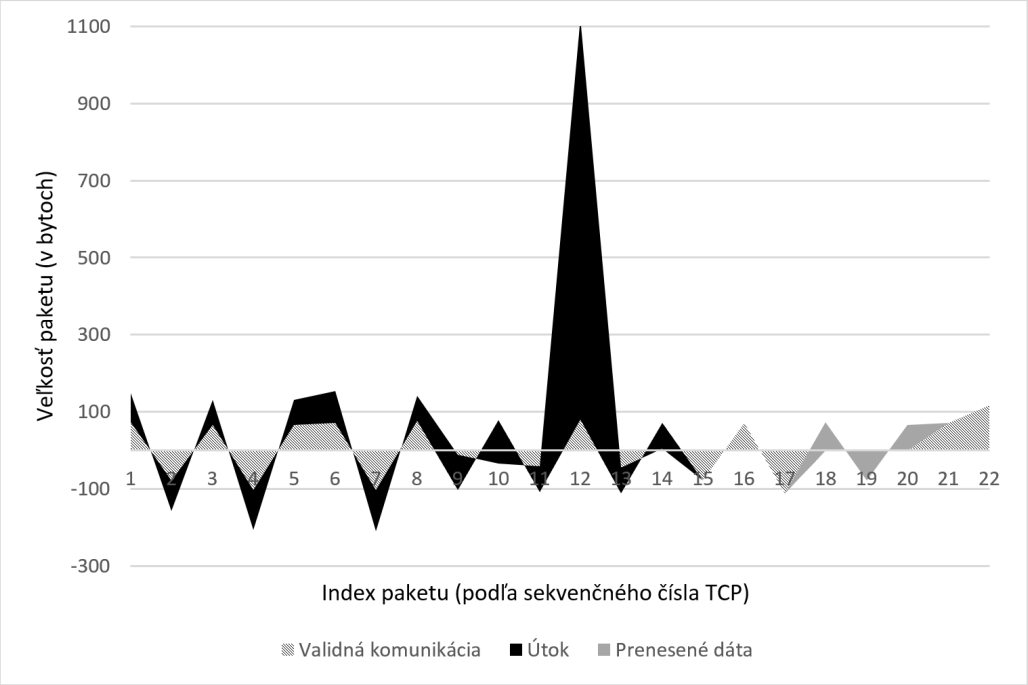
Na obrázku 5.11 je zobrazené distribučné rozloženie polynomiálnych aproximácii (1. a 2. koeficient) pre znázornenie pravdepodobnostného ohodnotenia daného rysu voči klasifikačným triedam. Červenou farbou sú zobrazené útoky a modrou farbou validné komunikácie.

5.5 Case Studies

V rámci experimentov s definovanými metrikami a metódami analýzy je možné ukázať vlastnosti komunikácie analýzou konkrétneho prípadu validnej komunikácie a útoku v tom istom (laboratórnom) prostredí. V rámci publikácie *Detection of Network Buffer Overflow Attacks: a Case Study* [14] boli tieto vlastnosti ukázané na dvoch konkrétnych útokoch:

- Buffer overflow útok na CesarFTP v0.99g (zraniteľnosť CVE-2006-2961);
- Buffer overflow útok na FreeSSHd v1.0.9 (zraniteľnosť CVE-2006-2407).

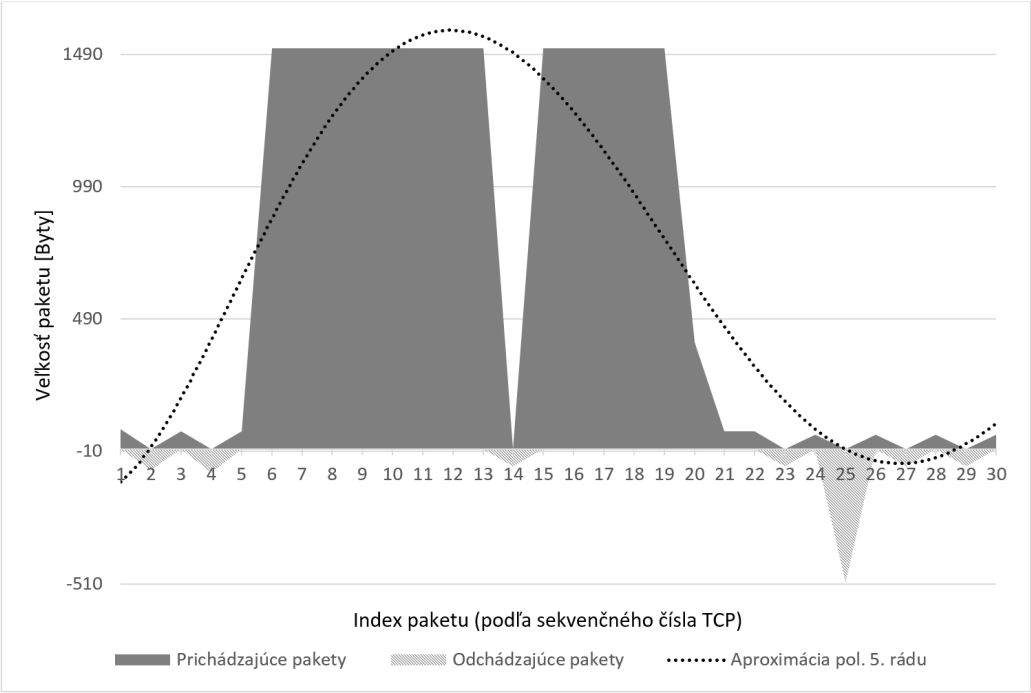
Oba útoky boli vedené pomocou frameworku *Metasploit*, ktorý pre uvedené zraniteľností obsahuje verejné exploity[6]. Na obrázku 5.12 je zobrazený útok na zraniteľnú sieťovú službu FTP spolu so simulovanou validnou komunikáciou (simulácia bola vykonaná bežným pripojením na službu a listovaním v adresári FTP). Smer paketov je zobrazený kladnou/zápornou hodnotou údajov (smer od klienta na FTP server je kladná – nad osou x a opačný smer je vyobrazený zápornou hodnotou – pod osou x). Hodnoty grafu predstavujú veľkosť prenesených dát (paketov) a to v indexovej rovine (definičný obor je tvorený indexom paketu). Vrchol (12-ty paket) je paket, ktorý obsahuje payload exploitu buffer overflow zraniteľnosti.



Obr. 5.12: Graf zobrazujúci veľkosť paketu v definičnom obore indexu paketu pre útok a validnú komunikáciu na FTP službu.

V rámci protokolu FTP (podľa RFP765 [106]) je komunikácia definovaná v následných krokoch s naviazaním spojenia a následne procedúrou prihlásenia (po 9. paket) a riadiacimi príkazmi (10. až 15. paket), kde 12. paket je vytvorenie adresára, ktoré sa vloží do pripraveného bufferu. V prípade validnej komunikácie je poslaný krátky reťazec "`MKD `", ktorý je konkatenovaný s reťazcom mena adresára a v prípade útoku je tento reťazec nahradený payloadom, ktorý pretečie pripravený buffer a návratovú hodnotu do pôvodnej funkcie, čo spôsobí skok programu na zadaný reťazec namiesto na pozíciu inštrukcií pôvodného programu. V prípade, že detekčný systém je naučený na postupnosť príkazov FTP protokolu a v komunikácii sa vyskytne paket (prípadne pakety), ktoré sú veľkosťou mimo obvyklý rozsah, je možné s vysokou pravdepodobnosťou zachytiť prebiehajúci útok.

[6]Exploit pre CesarFTP 0.99g je možné nájsť taktiež v databáze Exploit-DB: `https://www.exploit-db.com/exploits/1906/` a exploit pre FreeSSHd 1.0.9 na adrese `https://www.exploit-db.com/exploits/1787/`, ako aj obe zraniteľné verzie programov.



Obr. 5.13: Graf zobrazujúci veľkosť paketu v definičnom obore indexu paketu pre útok na SSH službu.

Na obrázku 5.13 je zobrazený útok na SSH službu, ktorá v priebehu niekoľkých paketov pred prvým vstupom od užívateľa, ktorým je autentizácia ustanoví spojenie (napr. vzájomná dohoda na použitých algoritmoch, výmena kľúčov, typ autentizácie apod.), nasleduje prihlásenie užívateľa a otvorenie spojenia v prípade úspešnej autentizácie. V prípade zobrazeného útoku je napadnutá funkcia služby SSH, ktorá prijíma od klienta pripojujúceho sa k službe návrh podporujúcich algoritmov na výmenu kľúčov (pre ustanovenie zabezpečeného kanálu). Táto funkcia očakáva na vstupe reťazec, ktorý špecifikuje podporované metódy výmeny kľúčov klientskej aplikácie. V danej verzii aplikácie FreeSSHd v1.0.9 sa nachádzala chyba neošetrenia veľkosti vstupného reťazca (kontrola vstupu na veľkosť podľa veľkosti alokovaného bufferu), čím je možné buffer preplniť (spôsobiť pretečenie na zásobníku, tzv. buffer overflow) a získať úplnú kontrolu nad programom i systémom (začnú sa vykonávať príkazy zo vstupu útočníka).

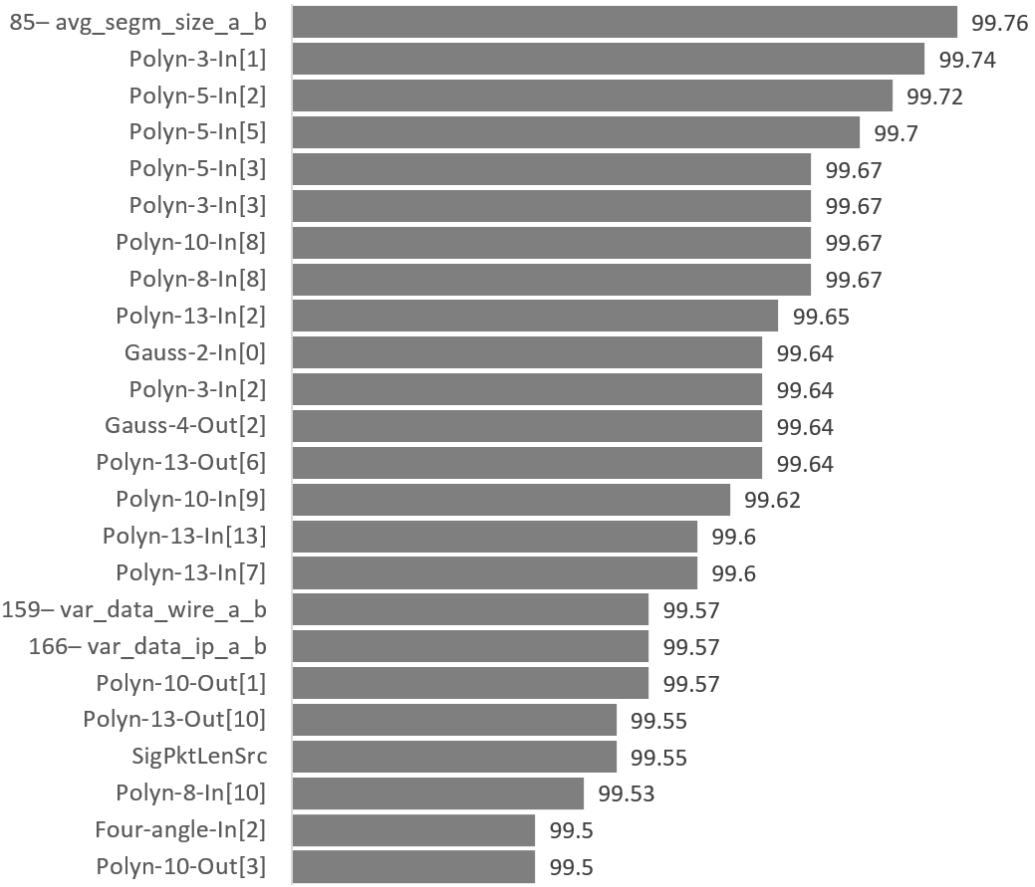
5.6 Zhrnutie experimentov

Navrhnutý systém bol experimentálne čiastočne implementovaný v podobe skriptov a modelov nástroja RapidMiner [112]. Vzhľadom na potreby vstupnej dátovej sady sieťového toku, ktorá by obsahovala dostatočné množstvo útokov a validnej komunikácie, ktorá by zodpovedala parametrom reálnej komunikácie, bolo vytvorené laboratórne prostredie a simulované útoky na služby honeypot systémov. Tieto dáta boli použité pri experimentoch s klasifikačnými metódami, ale použitá dátová sada nemala potrebné kvalitatívne parametre pre uspokojivé výsledky experimentov. Použité honeypoty boli z tohto dôvodu nasadené v sieti Vysokého učení technického v Brně pre zber potrebných záznamov útokov a sieťová sonda pre zber reálnych validných komunikácií. Tento projekt ale nepriniesol dostatočnú diver-

zitu útokov a bol ukončený. Použité honeypot systémy ďalej pokračovali v zbere útokov, ktoré sa používali pre identifikáciu infikovaných počítačov v sieti (a následné upovedomenie majiteľa o detegovanej infekcii systému zo strany Kolejí a menz VUT v Brně). Pre ďalšie experimenty boli použité verejne dostupné databázy a navrhnutá behaviorálna signatúra bola použitá pri klasifikácii rôznymi metódami ako napr. rozhodovacím stromom alebo bayesovským klasifikátorom pre overenie jej detekčnej schopnosti. Ďalej bol experimentami potvrdený predpoklad, že aproximácia priebehu spojenia ako simulácia správania analyzovaného spojenia a jeho aktérov bude vo výsledkoch experimentov dosahovať vysokú úspešnosť klasifikácie. Najlepšie dosiahnuté výsledky týchto experimentov sú zhrnuté v nasledujúcej kapitole.

5.6.1 Dosiahnuté výsledky

V práci bolo vykonaných niekoľko experimentov nad dátami vytvorenými v laboratórnom prostredí a dátami z verejných dátových sád, z ktorých pre finálne experimenty bola vybraná dátová sada CDX 2009. Výsledky testov v laboratórnom prostredí neboli z dôvodu nízkej kvality nazbieraných dát uspokojivé, naopak výsledky s dátami CDX 2009 sú blízke výsledkom dosiahnutým v obdobných výskumných prácach.



Obr. 5.14: Prehľad najlepších metrík a diskriminátorov zoradených podľa celkovej presnosti klasifikácie (nad 99,50 %).

Pri experimentoch s oboma vstupnými dátovými sadami bolo použitých niekoľko klasifikačných algoritmov. Experimenty s algoritmom rozhodovacieho stromu dosiahli klasifikačnú presnosť **99,76 %** s celkovým F-score **83 %** a najlepší bayesovský klasifikátor **99,83 %** presnosť s celkovým F-score **87,81 %**. Najlepšie rysy (a tým i najlepšie metriky) pre detekciu buffer overflow útokov zo sieťového toku, vyšli aproximačné funkcie. V najlepších 24 rysoch podľa celkovej presnosti klasifikácie nad 99,5 % (zobrazené na obrázku 5.14) bayesovským klasifikátorom sa nachádzajú polynómy až 17 krát (najviac 13. stupňa, ktorý má zastúpenie 5 krát). Vzhľadom na smer komunikácie je v rámci aproximácie polynómami vo väčšej miere zastúpená aproximácia prichádzajúcej (smer od zdroja k cieľu) komunikácie a to z dôvodu prítomnosti veľkého paketu nesúceho obsah útoku na prijímajúci buffer služby.

Kapitola 6

Záver

V tejto práci bol predstavený stručný úvod do problematiky sieťovej bezpečnosti a základné rozdelenie sieťových bezpečnostných technológií. V úvode boli vytýčené ciele práce a požiadavky na návrh autonómneho detekčného systému so zameraním na behaviorálnu analýzu sieťového toku a rozpoznanie anomálií. Hlavným cieľom práce je vytvorenie konceptu detekčného systému, ktorý pomocou definovaných metrík redukuje sieťový tok na signatúry spojení s dôrazom na autonómnosť systému pomocou vytvorenia expertnej znalosti honeypot systému, ako učiteľa a nezávislosť na technologických aspektoch analyzovaných dát (ako napr. šifrovanie, použité protokoly, technológie, či prostredie).

6.1 Prínos práce

Prínosy práce je možné sumarizovať pomocou definovaného cieľa práce v úvode kapitoly, ale celkový prínos je možné rozdeliť do niekoľkých bodov, ktoré lepšie poskytnú prehľad o dosiahnutých výsledkoch.

- Jedným z hlavných prínosov práce je zakomponovanie **behaviorálnej charakteristiky spojenia** na zlepšenie detekcie špecifických sieťových útokov (so zameraním na buffer overflow útoky). Tento cieľ bol dosiahnutý vytvorením špecifikácie signatúry spojenia, ktorá obsahuje koeficienty aproximácií priebehu spojenia variabilných atribútov charakteristiky sieťového toku. Celkové výsledky naznačujú vysokú mieru klasifikácie týchto útokov práve pomocou behaviorálnej časti signatúry až do 99,83 % presnosti klasifikácie.
- Práca poskytuje **ucelený formalizovaný prístup** k definícii charakteristiky sieťového toku, metrík ako funkcií, ktoré z definovanej charakteristiky vytvárajú signatúru spojenia, modelu detekčného systému a to s dôraznosťou na exaktnosť definícií a rozšíriteľnosť systému o ďalšie funkcie a metódy. Tento formálny model obsahuje všetky atribúty sieťového toku a je možné nad ním vystavať ďalšie detekčné mechanizmy, podobne je možná redukcia atribútov, prípadne metrík pre minimalizáciu veľkosti signatúry a zrýchlenie klasifikačných algoritmov.
- Práca prináša ďalší pohľad pre detekciu komplikovanejších útokov pomocou **kontextu spojenia**, ktorým je možné modelovať správanie analyzovaných uzlov (systémov) v sieti počas určitého časového obdobia, väzby medzi sieťovými prvkami a pozorovať tak vzory správania analyzovaného spojenia alebo systému.

- Zakomponovanie autonómnosti ako prvku detekčného systému pomocou nasadeného honeypot systému. Vytvorenie väzby expertnej znalosti honeypot systému v roli učiteľa klasifikačných algoritmov vytvára autonómnosť systému pri detekcii i neznámych útokov a možnosť samostatného učenia (v zmysle bez zásahu človeka) na základe poznatkov zbieraných z honeypot systémov.
- Počas práce s vedeckými prácami a publikáciami o sieťových detekčných systémoch a metódach bola vypracovaná štúdia (pozri kap. 2), ktorá sumarizuje dosiahnuté výsledky za posledných 20 rokov s poukázaním na problémy, ktorým čelia výskumné práce v tejto oblasti.

6.1.1 Vedecký prínos

Popisované výsledky práce nadväzujú a rozširujú aktuálny stav bádania v oblasti sieťovej bezpečnosti o behaviorálne aspekty komunikácie pomocou aproximačných funkcií a tým zlepšujú detekčné schopnosti pre špecifické útoky. Práca ďalej prináša nové využitie honeypot systémov, ktoré vystupujú ako expertné systémy zahrnuté do kontinuálneho procesu automatického učenia detekčných algoritmov, čím zvyšujú autonómnosť celého systému. Jedným z prínosov je i vytvorenie formalizovaného prístupu k definícii charakteristiky spojenia, metrík a behaviorálnej signatúry, na ktorom je možné stavať ďalšie rozšírenia detekčných techník.

Všetky popisované prínosy tejto práce sú dielom autora tejto práce. Na vytvorení testovacieho prostredia a spolupráci v oblasti experimentov sa podieľali ďalší spolupracovníci v rámci projektu *Automatizované zpracování útoků* (FR-TI1/037 financovaný MPO ČR).

6.2 Zhodnotenie

Koncept detekčného systému s využitím aproximácií priebehu spojení je podľa uvedených výsledkov experimentov zaujímavým riešením a to i z pohľadu konceptu honeypotov, ktoré nie sú bežne používanou technológiou v detekčných systémoch.

Najlepšie uvedené výsledky obdobných výskumných projektov (uvedené v kap. 2.2.4) dosiahli systémy *AHM-NID* (99,60 %), alebo *Octopus-IIDS* (97,40 %), ale len s analýzou obsahov paketov, ktorá nie je vždy možná. Medzi riešenia analyzujúce atribúty paketov je možné zaradiť výskumnú prácu Moore [90, 89, 10], ktorý dosiahol 95 % presnosť nad vlastnou dátovou sadou. Navrhnutý systém pomocou behaviorálnej signatúry dosiahol celkovú presnosť klasifikácie 99,83 %. Nie je účelom tejto práce tvrdiť, že systém je v účinnosti detekcie najlepší alebo, že navrhnutý koncept je ten správny, ale dosiahnuté výsledky poukazujú na zaujímavý prístup analýzy sieťového toku.

6.3 Budúcnosť

Práca predstavuje koncept detekčného systému, ktorý predstavuje základ pre ďalšie výskumné práce, rozšírenia detekčných techník, zlepšenie modelu z pohľadu optimalizácie apod. Hlavné oblasti, ktoré budú v budúcnosti predmetom ďalšieho výskumu je možné zhrnúť do nasledujúcich bodov.

UDP komunikácie. Aktuálny koncept systému pracuje nad TCP/IPv4 architektúrou. V minulosti ale boli zaznamenané útoky na protokoly pracujúce nad UDP. Príkladom môže

byť zraniteľná služba *Sentinel LM* v programe *Sentinel License Manager 7.2.0.2*, ktorá umožňuje útočníkovi spustiť vlastný vytvorený kód na vzdialenom systéme pomocou zaslania veľkého dátového toku na UDP port 5093 (pozri CVE-2005-0353). Sieťový tok na UDP protokole je problematický pri identifikácii začiatku a konca spojenia, ale aj charakteristika spojenia môže byť pri tej istej službe vždy iný, keďže k doručeniu niektorých paketov nemusí počas spojenia dôjsť. UDP spojenia sú väčšinou služby, ako VoIP alebo streamovacie služby (video, hudba), ktoré môžu trvať dlhý časový interval (prípadne môžu byť z pohľadu detekčného systému nekonečné), čo vytvára novú požiadavku detekčného systému a to analýza spojení počas ich priebehu.

Optimalizácia signatúry. Vytvorená signatúra spojenia obsahuje definovanými sieťovými metrikami spracovanú kompletnú charakteristiku spojenia. Tá bola pri rozšírení modelu na TCP/IPv4 architektúru zmenšená o niektoré nepotrebné atribúty TCP a IP paketov (pozri kapitolu 3.4.4). I tak ale signatúra stále obsahuje 657 hodnôt metrík a dosahuje veľkosť 1,9 kB na spojenie, čo je v prípade nasadenia systému na vysoko-rýchlostné siete veľké číslo. Pri optimalizácii (čo najväčšom zmenšení výslednej signatúry) by ale nemalo dochádzať k znižovaniu kvality signatúry (strate informácií). Jednou z možností je i optimalizácia požitých algoritmov, spracovanie tokov alebo aproximačných algoritmov presunom do špecializovaného hardware.

Databáza útokov. Vzhľadom na neexistenciu vhodnej databázy útokov, ktorá by spĺňala podmienky, ktoré sú na ňu kladené klasifikačnými experimentami, vznikla po vytvorení laboratórneho prostredia snaha o vytvorenie komplexnej verejnej databázy, ktorá by slúžila všetkým výskumných skupinám a organizáciám pre experimenty s detekčnými nástrojmi a metódami. Táto databáza by mala byť automatizovaná s možnosťou dodania zraniteľného programu a exploitu pre simulovanie útoku a zber dát. Využitím tejto databázy je možné v rámci podmienok stanoviť napr. povinnosť výskumníka zverejniť úplné dosiahnuté výsledky tak, aby bolo možné porovnať účinnosť zvolených metód verejnosťou.

Neukončené spojenia. Jedným z problematických súčastí analýzy sieťového toku v reálnom alebo skoro reálnom čase, sú neukončené spojenia. Podľa dostupnej literatúry (RFC 793 [105]) je spojenie ukončené až po 5-tich minútach v prípade, že v tomto čase nedôjde k prijatiu ďalšieho paketu. Tento časový interval (i v prípade, že by bol menší) môže spôsobiť veľké výkonnostné problémy udržiavania celých spojení v pamäti analyzátora a zdržovanie analýzy a následného vyhodnotenia po dobu väčšiu ako 5 minút. To platí i pre neukončené spojenia. Jedným z riešení je vytvorenie kontinuálneho procesu, ktorý po veľmi krátkom intervale (v závislosti na danej sieti, rádovo v sekundách) spúšťa proces vytvorenia charakteristiky a signatúry spojenia, ktoré budú zaradené do analýzy a v prípade, že príde ďalší paket, ktorý patrí do daného spojenia, bude charakteristika i daná signatúra aktualizovaná a proces analýzy reštartovaný. Tento prístup, ani iné prístupy pre vyriešenie tohoto problému neboli doteraz otestované.

Ďalší výskum. Ďalší výskum nad navrhnutým konceptom detekčného nástroja sa môže vzťahovať na využitie, prípadne úpravu lokálneho a globálneho kontextu spojenia, zdrojového a cieľového systému pre lepšiu detekciu pokročilých útokov. S tým súvisia abstraktnejšie úrovne detekčných metód, ako napríklad korelačné pravidlá, či modelovanie psychologického profilu systémov (rozpoznanie útoku na základe zmeny správania človeka). V práci

ďalej chýba zapojenie kontextu do experimentov, či možnosť analyzovať špecifické protokoly, ktoré sú na vzostupe obdobných výskumných projektov (napr. protokoly riadiacich systémov ICS). Jednou z ďalších oblastí, ktoré stoja za pozornosť v budúcnosti projektu je pridanie možnosti aktívneho zabránenia prienikov do siete, prípadne zefektívnenie detekcie, zameranie sa na abstraktnejšie metódy (korelácie, modelovanie vzťahov apod).

Vyriešenie týchto oblastí je síce nad rámec tejto práce, ale niektoré sú v dobe odovzdávania tejto práce v štádiu riešenia. Uvedený koncept systému je aktuálne pripravovaný pre transfer do praxe a jeho úspechy, či neúspechy pri detekcii sieťových útokov v reálnom prostredí ukáže budúcnosť.

Literatúra

[1] Adachi, Y.; Oyama, Y.: Malware analysis system using process-level virtualization. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, IEEE, 2009, s. 550–556.

[2] Aha, D. W.; Kibler, D.; Albert, M. K.: Instance-based learning algorithms. *Machine learning*, ročník 6, č. 1, 1991: s. 37–66.

[3] Alberdi, I.; Alata, E.; Nicomette, V.; a kol.: Shark: Spy honeypot with advanced redirection kit. In *IEEE Workshop on Monitoring, Attack Detection and Mitigation (MonAM 07)*, 2007, s. 47–52.

[4] Allman, M.; Paxson, V.; Blanton, E.: TCP congestion control. Technická správa, IETF, 2009.

[5] Alosefer, Y.; Rana, O.: Honeyware: a web-based low interaction client honeypot. In *Software Testing, Verification, and Validation Workshops (ICSTW), 2010 Third International Conference on*, IEEE, 2010, s. 410–417.

[6] Anagnostakis, K. G.; Sidiroglou, S.; Akritidis, P.; a kol.: Detecting Targeted Attacks Using Shadow Honeypots. In *Usenix Security*, 2005, s. 129–144.

[7] Anderson, D.; Lunt, T. F.; Javitz, H.; a kol.: *Detecting unusual program behavior using the statistical component of the Next-generation Intrusion Detection Expert System (NIDES)*. SRI International, Computer Science Laboratory, 1995.

[8] Angiulli, F.; Argento, L.; Furfaro, A.: PCkAD: an unsupervised intrusion detection technique exploiting within payload n-gram location distribution. *arXiv preprint arXiv:1412.3664*, 2014.

[9] Ariu, D.; Tronci, R.; Giacinto, G.: HMMPayl: An intrusion detection system based on Hidden Markov Models. *computers & security*, ročník 30, č. 4, 2011: s. 221–241.

[10] Auld, T.; Moore, A. W.; Gull, S. F.: Bayesian neural networks for internet traffic classification. *Neural Networks, IEEE Transactions on*, ročník 18, č. 1, 2007: s. 223–239.

[11] Auld, T.; Moore, A. W.; Gull, S. F.: Bayesian neural networks for internet traffic classification. *Neural Networks, IEEE Transactions on*, ročník 18, č. 1, 2007: s. 223–239.

[12] Baek, S.-I.; Kim, W.-S.; Koo, T.-M.; a kol.: Inspection of defect on LCD panel using polynomial approximation. In *TENCON 2004. 2004 IEEE Region 10 Conference*, IEEE, 2004, s. 235–238.

[13] Barabas, M.; Homoliak, I.; Drozd, M.; a kol.: Automated Malware Detection Based on Novel Network Behavioral Signatures. *International Journal of Engineering and Technology*, ročník 5, č. 2, 2013: str. 249.

[14] Barabas, M.; Homoliak, I.; Kacic, M.; a kol.: Detection of network buffer overflow attacks: A case study. In *Security Technology (ICCST), 2013 47th International Carnahan Conference on*, IEEE, 2013, s. 1–4.

[15] Barbara, D.; Wu, N.; Jajodia, S.: Detecting Novel Network Intrusions Using Bayes Estimators. In *SDM*, SIAM, 2001, s. 1–17.

[16] Barrantes, E. G.; Ackley, D. H.; Palmer, T. S.; a kol.: Randomized instruction set emulation to disrupt binary code injection attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*, ACM, 2003, s. 281–289.

[17] Bhatkar, S.; DuVarney, D. C.; Sekar, R.: Address Obfuscation: An Efficient Approach to Combat a Broad Range of Memory Error Exploits. In *USENIX Security*, ročník 3, 2003, s. 105–120.

[18] Bhatkar, S.; DuVarney, D. C.; Sekar, R.: Efficient Techniques for Comprehensive Protection from Memory Error Exploits. In *Usenix Security*, 2005, s. 271–286.

[19] Bolle, R. M.; Cooper, D. B.: Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ročník PAMI-6, č. 4, 1984: s. 418–429.

[20] Bolon-Canedo, V.; Sanchez-Marono, N.; Alonso-Betanzos, A.: Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset. *Expert Systems with Applications*, ročník 38, č. 5, 2011: s. 5947–5957.

[21] Bolzoni, D.; Zambon, E.; Etalle, S.; a kol.: Poseidon: A 2-tier anomaly-based intrusion detection system. *arXiv preprint cs/0511043*, 2005.

[22] Bonner, L.: Cyber Risk: How the 2011 Sony Data Breach and the Need for Cyber Risk Insurance Policies Should Direct the Federal Response to Rising Data Breaches. *Wash. UJL & Pol'y*, ročník 40, 2012: str. 257.

[23] Boser, B. E.; Guyon, I. M.; Vapnik, V. N.: A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, ACM, 1992, s. 144–152.

[24] Bramer, M.: Pre-pruning classification trees to reduce overfitting in noisy domains. In *Intelligent Data Engineering and Automated Learning – IDEAL 2002*, Springer, 2002, s. 7–12.

[25] Breiman, L.: Random forests. *Machine learning*, ročník 45, č. 1, 2001: s. 5–32.

[26] Bringer, M. L.; Chelmecki, C. A.; Fujinoki, H.: A survey: Recent advances and future trends in honeypot research. *International Journal*, ročník 4, 2012.

[27] Brownlee, Nevil and Mills, Cyndi and Ruth, Greg: Traffic Flow Measurement: Architecture. RFC 2722, IETF, October 1999.
URL `https://tools.ietf.org/html/rfc2722.html`

[28] Burges, C. J.: A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, ročník 2, č. 2, 1998: s. 121–167.

[29] Cao, Y.; Wu, J.: Projective ART for clustering data sets in high dimensional spaces. *Neural networks*, ročník 15, č. 1, 2002: s. 105–120.

[30] Christodorescu, M.; Jha, S.; Maughan, D.; a kol.: *Malware Detection*, ročník 27. Springer Science & Business Media, 2007.

[31] C.I.A. World Factbook. 2015.
URL `https://www.cia.gov/library/publications/resources/the-world-factbook/rankorder/2153rank.html`

[32] Cisco Systems, I.: NetFlow. 2011.
URL `http://www.cisco.com/go/netflow`

[33] Cisco Systems, I.: Cisco 2014 Annual Security Report. 2014.

[34] Clark, P.: Machine learning: techniques and recent developments. 1990.

[35] Cooper, G. F.; Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, ročník 9, č. 4, 1992: s. 309–347.

[36] Cortes, C.; Vapnik, V.: Machine learning. *Support-vector networks, journal*, ročník 20, 1995: s. 273–297.

[37] Cortes, C.; Vapnik, V.: Support-vector networks. *Machine learning*, ročník 20, č. 3, 1995: s. 273–297.

[38] Cortes, C.; Vapnik, V.: Support-vector networks. *Machine learning*, ročník 20, č. 3, 1995: s. 273–297.

[39] Cowan, C.; Wagle, P.; Pu, C.; a kol.: Buffer overflows: Attacks and defenses for the vulnerability of the decade. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, ročník 2, IEEE, 2000, s. 119–129.

[40] Crandall, J. R.; Su, Z.; Wu, S. F.; a kol.: On deriving unknown vulnerabilities from zero-day polymorphic and metamorphic worm exploits. In *Proceedings of the 12th ACM conference on Computer and communications security*, ACM, 2005, s. 235–248.

[41] Cyberwar: War in the Fifth Domain.
URL `http://www.economist.com/node/16478792`

[42] Daneels, A.; Salter, W.: What is SCADA? 1999.

[43] De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D. L.: The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, ročník 50, č. 1, 2000: s. 1–18.

[44] Denning, D. E.; Neumann, P. G.: Requirements and model for IDES–a real-time intrusion detection expert system. *Document A005, SRI International*, ročník 333, 1985.

[45] Dressler, F.; Jaegers, W.; German, R.: Flow-based worm detection using correlated honeypot logs. In *Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference*, VDE, 2007, s. 1–6.

[46] Drozd, M.; Barabas, M.; Gregr, M.; a kol.: Buffer overflow attacks data acquisition. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, ročník 2, IEEE, 2011, s. 775–779.

[47] Du, W.; Zhan, Z.: Building decision tree classifier on private data. In *Proceedings of the IEEE international conference on Privacy, security and data mining-Volume 14*, Australian Computer Society, Inc., 2002, s. 1–8.

[48] Elkan, C.: Results of the KDD'99 classifier learning. *ACM SIGKDD Explorations Newsletter*, ročník 1, č. 2, 2000: s. 63–64.

[49] Ertoz, L.; Eilertson, E.; Lazarevic, A.; a kol.: Detection and summarization of novel network attacks using data mining. *Minnesota INtrusion Detection System (MINDS) Technical Report*, 2003.

[50] Finkle, J.; Shalal-Esa, A.: Exclusive: Hackers breached US defense contractors. *Reuters. Accessed*, ročník 20120517, 2011.

[51] Freund, Y.; Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, ročník 55, č. 1, 1997: s. 119–139.

[52] Gartner's hype cycle special report for 2014.
URL `http://www.gartner.com/newsroom/id/2819918`

[53] Ghorbani, A. A.; Lu, W.; Tavallaee, M.: Evaluation Criteria. In *Network Intrusion Detection and Prevention*, Springer, 2010, s. 161–183.

[54] Gil, T. M.; Poletto, M.: MULTOPS: a data-structure for bandwidth attack detection. In *USENIX Security Symposium*, 2001, s. 23–38.

[55] Goutte, C.; Gaussier, E.: A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *Advances in information retrieval*, Springer, 2005, s. 345–359.

[56] Goyal, R.; Sharma, S.; Bevinakoppa, S.; a kol.: Obfuscation of stuxnet and flame malware. *Latest Trends in Applied Informatics and Computing*, 2012.

[57] Gunn, S. R.; a kol.: Support vector machines for classification and regression. *ISIS technical report*, ročník 14, 1998.

[58] Haijun, X.; Fang, P.; Ling, W.; a kol.: Ad hoc-based feature selection and support vector machine classifier for intrusion detection. In *Grey Systems and Intelligent Services, 2007. GSIS 2007. IEEE International Conference on*, IEEE, 2007, s. 1117–1121.

[59] Halliday, J.: Epsilon email hack: millions of customers' details stolen.

[60] Han, J.; Kamber, M.: Data Mining: Concepts and Techniques. *University of Illinois at Urbana-Champaign,*, 2006.

[61] Haque, M. E.; Alkharobi, T. M.: Adaptive Hybrid Model for Network Intrusion Detection and Comparison among Machine Learning Algorithms. *International Journal of Machine Learning and Computing*, ročník 5, č. 1, 2015: str. 17.

[62] Helali, R. G. M.: Data mining based network intrusion detection system: A survey. In *Novel Algorithms and Techniques in Telecommunications and Networking*, Springer, 2010, s. 501–505.

[63] Herrero, Á.; Navarro, M.; Corchado, E.; a kol.: RT-MOVICAB-IDS: Addressing real-time intrusion detection. *Future Generation Computer Systems*, ročník 29, č. 1, 2013: s. 250–261.

[64] Hettich, S.; Bay, S.: Kdd cup 1999 data. *The UCI KD Archive, Irvine, CA: University of California, Department of Information and Computer Science*, 1999.

[65] Hirschberg, D. S.: Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM)*, ročník 24, č. 4, 1977: s. 664–675.

[66] Hsu, F.-H.; Chiueh, T.-c.: CTCP: a transparent centralized tcp/ip architecture for network security. In *Computer Security Applications Conference, 2004. 20th Annual*, IEEE, 2004, s. 335–344.

[67] Idika, N.; Mathur, A. P.: A survey of malware detection techniques. *Purdue University*, ročník 48, 2007.

[68] ISACA: Advanced Persistent Threat Awareness Study Results. Technická správa, ISACA, 2014.

[69] Jacobson, V.; Leres, C.; McCanne, S.: libpcap, Lawrence Berkeley Laboratory, Berkeley, CA. *Initial public release June*, 1994.

[70] Jauregui, J.: Principal component analysis with linear algebra. 2012.

[71] Jordan, T.; Taylor, P. A.: *Hacktivism and cyberwars: rebels with a cause?* Psychology Press, 2004.

[72] Kantardzic, M.: *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[73] Khosravifar, B.; Bentahar, J.: An experience improving intrusion detection systems false alarm ratio by using honeypot. In *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, IEEE, 2008, s. 997–1004.

[74] Kim, H.-A.; Karp, B.: Autograph: Toward Automated, Distributed Worm Signature Detection. In *USENIX security symposium*, ročník 286, San Diego, CA, 2004.

[75] Kreibich, C.; Crowcroft, J.: Honeycomb: creating intrusion detection signatures using honeypots. *ACM SIGCOMM Computer Communication Review*, ročník 34, č. 1, 2004: s. 51–56.

[76] Kushner, D.: The real story of stuxnet. *IEEE Spectrum*, ročník 50, č. 3, 2013: s. 48–53.

[77] Kushner, D.: The real story of stuxnet. *IEEE Spectrum*, ročník 50, č. 3, 2013: s. 48–53.

[78] Lee, W.; Stolfo, S. J.: A framework for constructing features and models for intrusion detection systems. *ACM transactions on Information and system security (TiSSEC)*, ročník 3, č. 4, 2000: s. 227–261.

[79] Liang, Z.; Sekar, R.: Automatic generation of buffer overflow attack signatures: An approach based on program behavior models. In *Computer Security Applications Conference, 21st Annual*, IEEE, 2005, s. 10–pp.

[80] Madhukar, A.; Williamson, C.: A longitudinal study of P2P traffic classification. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*, IEEE, 2006, s. 179–188.

[81] Mafra, P. M.; Moll, V.; da Silva Fraga, J.; a kol.: Octopus-IIDS: An anomaly based intelligent intrusion detection system. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, IEEE, 2010, s. 405–410.

[82] Mahoney, M. V.; Chan, P. K.: Learning nonstationary models of normal network traffic for detecting novel attacks. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2002, s. 376–385.

[83] Mahoney, M. V.; Chan, P. K.: An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In *Recent Advances in Intrusion Detection*, Springer, 2003, s. 220–237.

[84] Mairh, A.; Barik, D.; Verma, K.; a kol.: Honeypot in network security: a survey. In *Proceedings of the 2011 International Conference on Communication, Computing & Security*, ACM, 2011, s. 600–605.

[85] Maria Garnaeva, a. s.: Kaspersky Security Bulletin 2014.
URL `https://securelist.com/analysis/kaspersky-security-bulletin/68010/kaspersky-security-bulletin-2014-overall-statistics-for-2014/`

[86] Marill, T.; Green, D. M.: On the effectiveness of receptors in recognition systems. *Information Theory, IEEE Transactions on*, ročník 9, č. 1, 1963: s. 11–17.

[87] Measuring the Information Society Report 2014, Executive Summary. 2014 ITU.
URL `https://www.itu.int/dms_pub/itu-d/opb/ind/D-IND-ICTOI-2014-SUM-PDF-E.pdf`

[88] Miller, C.; Valasek, C.: Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015.

[89] Moore, A.; Zuev, D.; Crogan, M.: *Discriminators for use in flow-based classification.* Queen Mary and Westfield College, Department of Computer Science, 2005.

[90] Moore, A. W.; Zuev, D.: Internet traffic classification using bayesian analysis techniques. In *ACM SIGMETRICS Performance Evaluation Review*, ročník 33, ACM, 2005, s. 50–60.

[91] Moore, D.; Shannon, C.; Voelker, G. M.; a kol.: Internet quarantine: Requirements for containing self-propagating code. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, ročník 3, IEEE, 2003, s. 1901–1910.

[92] Newsome, J.; Song, D.: Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software. 2005.

[93] Nguyen, T. T.; Armitage, G.: A survey of techniques for internet traffic classification using machine learning. *Communications Surveys & Tutorials, IEEE*, ročník 10, č. 4, 2008: s. 56–76.

[94] NoAH FP6 EU Project. 2008.
URL `http://www.fp6-noah.org/index.html`

[95] Norton-Taylor, R.: Titan Rain: How Chinese Hackers Targeted Whitehall. *The Guardian*, ročník 5, 2007.

[96] Oja, E.; Kaski, S.: *Kohonen maps*. Elsevier, 1999.

[97] Panda, M.; Abraham, A.; Patra, M. R.: A hybrid intelligent approach for network intrusion detection. *Procedia Engineering*, ročník 30, 2012: s. 1–9.

[98] Pasupulati, A.; Coit, J.; Levitt, K.; a kol.: Buttercup: On network-based detection of polymorphic buffer overflow vulnerabilities. In *Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP*, ročník 1, IEEE, 2004, s. 235–248.

[99] Paxson, V.: Bro: a system for detecting network intruders in real-time. *Computer networks*, ročník 31, č. 23, 1999: s. 2435–2463.

[100] Perdisci, R.; Ariu, D.; Fogla, P.; a kol.: McPAD: A multiple classifier system for accurate payload-based anomaly detection. *Computer networks*, ročník 53, č. 6, 2009: s. 864–881.

[101] Peter, E.; Schiller, T.: A practical guide to honeypots. *Washington Univerity*, 2011.

[102] Philips, W.; De Jonghe, G.: Data compression of ECG's by high-degree polynomial approximation. *Biomedical Engineering, IEEE Transactions on*, ročník 39, č. 4, 1992: s. 330–337.

[103] Porras, P. A.; Neumann, P. G.: EMERALD: Event monitoring enabling response to anomalous live disturbances. In *Proceedings of the 20th national information systems security conference*, 1997, s. 353–365.

[104] Portokalidis, G.; Slowinska, A.; Bos, H.: Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation. In *ACM SIGOPS Operating Systems Review*, ročník 40, ACM, 2006, s. 15–27.

[105] Postel, J.: Transmission Control Protocol. RFC 793, IETF, September 1981.
URL `https://tools.ietf.org/html/rfc793`

[106] Postel, J.; Reynolds, J.: File Transfer Protocol (FTP). RFC 765, IETF, June 1980.
URL `https://tools.ietf.org/html/rfc765`

[107] Provos, N.: Honeyd-a virtual honeypot daemon. In *10th DFN-CERT Workshop, Hamburg, Germany*, ročník 2, 2003, str. 4.

[108] Pudil, P.; Novovičová, J.; Kittler, J.: Floating search methods in feature selection. *Pattern recognition letters*, ročník 15, č. 11, 1994: s. 1119–1125.

[109] Qualys, I.: The laws of vulnerabilities: Six axioms for understanding risk. *Qualys White Paper*, 2006.

[110] Rajahalme, Jarno and Amante, Shane and Jiang, Sheng and Carpenter, Brian: IPv6 Flow Label Specification. RFC 3697, IETF, November 2011.
URL `https://tools.ietf.org/html/rfc3697.html`

[111] Ramakrishnan, K.; Floyd, S.; Black, D.: The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, IETF, September 2001.
URL `https://tools.ietf.org/html/rfc3168.html`

[112] RapidMiner.
URL `https://rapidminer.com/`

[113] "Red October"Diplomatic Cyber Attacks Investigation.
URL `https://securelist.com/analysis/publications/36740/red-october-diplomatic-cyber-attacks-investigation/`

[114] Rowe, N. C.; Custy, E. J.; Duong, B. T.: Defending cyberspace with fake honeypots. *Journal of Computers*, ročník 2, č. 2, 2007: s. 25–36.

[115] Sangster, B.; O'Connor, T.; Cook, T.; a kol.: Toward Instrumenting Network Warfare Competitions to Generate Labeled Datasets. In *CSET*, 2009.

[116] Scarfone, K.; Mell, P.: Guide to intrusion detection and prevention systems (idps). *NIST special publication*, ročník 800, č. 2007, 2007: str. 94.

[117] Secunia: Secunia Vulnerability Review 2015.

[118] Shin, S.; Lee, S.; Kim, H.; a kol.: Advanced probabilistic approach for network intrusion forecasting and detection. *Expert Systems with Applications*, ročník 40, č. 1, 2013: s. 315–322.

[119] Sokolova, M.; Japkowicz, N.; Szpakowicz, S.: Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *AI 2006: Advances in Artificial Intelligence*, Springer, 2006, s. 1015–1021.

[120] Spitzner, L.: *Honeypots: tracking hackers*, ročník 1. Addison-Wesley Reading, 2003.

[121] Stolfo, S. J.; Fan, W.; Lee, W.; a kol.: Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, ročník 2, IEEE, 2000, s. 130–144.

[122] Stouffer, K.; Falco, J.; Scarfone, K.: Guide to industrial control systems (ICS) security. *NIST special publication*, 2011: s. 800–82.

[123] Tankard, C.: Advanced Persistent threats and how to monitor and deter them. *Network security*, ročník 2011, č. 8, 2011: s. 16–19.

[124] Tavallaee, M.; Bagheri, E.; Lu, W.; a kol.: A detailed analysis of the KDD CUP 99 data set. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*, 2009.

[125] Tavallaee, M.; Stakhanova, N.; Ghorbani, A. A.: Toward credible evaluation of anomaly-based intrusion-detection methods. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, ročník 40, č. 5, 2010: s. 516–524.

[126] Teknos, M.: *Detekce honeypot systému v síti*. Diplomová práca, Fakulta informacnich technologii, VYSOKE UCENI TECHNICKE V BRNE, 2012.

[127] THE SNORT PROJECT. Snort, The OpenSource Network Intrusion Detection System.
URL `http://www.snort.org/`

[128] The Stuxnet worm: A cyber-missile aimed at Iran?
URL `http://www.economist.com/blogs/babbage/2010/09/stuxnet_worm`

[129] Thomas, C.; Sharma, V.; Balakrishnan, N.: Usefulness of DARPA dataset for intrusion detection system evaluation. In *SPIE Defense and Security Symposium*, International Society for Optics and Photonics, 2008, s. 69730G–69730G.

[130] Tzur-David, S.; Avissar, H.; Dolev, D.; a kol.: SPADE: Statistical Packet Acceptance Defense Engine. In *High Performance Switching and Routing (HPSR), 2010 International Conference on*, IEEE, 2010, s. 119–126.

[131] Utgoff, P. E.: Incremental induction of decision trees. *Machine learning*, ročník 4, č. 2, 1989: s. 161–186.

[132] Vigna, G.; Kemmerer, R.; a kol.: NetSTAT: A network-based intrusion detection approach. In *Computer Security Applications Conference, 1998. Proceedings. 14th Annual*, IEEE, 1998, s. 25–34.

[133] Wang, H. J.; Guo, C.; Simon, D. R.; a kol.: Shield: Vulnerability-driven network filters for preventing known vulnerability exploits. *ACM SIGCOMM Computer Communication Review*, ročník 34, č. 4, 2004: s. 193–204.

[134] Wang, K.; Parekh, J. J.; Stolfo, S. J.: Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Advances in Intrusion Detection*, Springer, 2006, s. 226–248.

[135] Wang, K.; Stolfo, S. J.: Anomalous payload-based network intrusion detection. In *Recent Advances in Intrusion Detection*, Springer, 2004, s. 203–222.

[136] Wang, K.; Stolfo, S. J.: Anomalous payload-based network intrusion detection. In *Recent Advances in Intrusion Detection*, Springer, 2004, s. 203–222.

[137] Wang, L.; Li, Z.; Chen, Y.; a kol.: Thwarting zero-day polymorphic worms with network-level length-based signature generation. *IEEE/ACM Transactions on Networking (TON)*, ročník 18, č. 1, 2010: s. 53–66.

[138] Wark, M.: Hacker Manifesto version 2.0. 2004.

[139] Weber, R. H.; Weber, R.: *Internet of Things.* New York: Springer, 1st Edition, 2010, iSBN: 978-3-642-11709-1.

[140] Whitney, A. W.: A direct method of nonparametric measurement selection. *Computers, IEEE Transactions on*, ročník 100, č. 9, 1971: s. 1100–1103.

Dodatok A

Publikácie

A.1 Publikácie relevantné k práci

BARABAS Maroš, HANÁČEK Petr, HOMOLIAK Ivan a KAČIC Matej. Detection of Network Buffer Overflow Attacks: A Case Study. In: The 47th Annual International Carnahan Conference on Security Technology. Mendellin: Institute of Electrical and Electronics Engineers, 2013, s. 128-131. ISBN 978-958-8790-65-7.

BARABAS Maroš, HOMOLIAK Ivan, DROZD Michal a HANÁČEK Petr. Automated Malware Detection Based on Novel Network Behavioral Signatures. International Journal of Engineering and Technology. Singapore: International Association of Computer Science and Information Technology, 2013, roč. 5, č. 2, s. 249-253. ISSN 1793-8236.

HOMOLIAK Ivan, BARABAS Maroš, CHMELAŘ Petr, DROZD Michal a HANÁČEK Petr. ASNM: Advanced Security Network Metrics for Attack Vector Description. In: Proceedings of the 2013 International Conference on Security & Management. Las Vegas: Computer Science Research, Education, and Applications Press, 2013, s. 350-358. ISBN 1-60132-259-3.

HOMOLIAK Ivan, BARABAS Maroš, CHMELAŘ Petr, DROZD Michal a HANÁČEK Petr. Advanced Security Network Metrics. Emerging Trends in ICT Security. Waltham: Elsevier Science, 2013, s. 187-201. ISBN 978-0-12-411474-6.

BARABAS Maroš, DROZD Michal a HANÁČEK Petr. Behavioral signature generation using shadow honeypot. World Academy of Science, Engineering and Technology. 2012, roč. 2012, č. 65, s. 829-833. ISSN 2010-376X.

DROZD Michal, BARABAS Maroš, GRÉGR Matěj a CHMELAŘ Petr. Buffer Overflow Attacks Data Acquisition. In: Proceedings of the 6th IEEE International Conference on IDAACS 2011. Praha: Institute of Electrical and Electronics Engineers, 2011, s. 775-779. ISBN 978-1-4577-1423-8.

A.2 Ďalšie vedecké publikácie

ANTAL Lukáš, BARABAS Maroš a HANÁČEK Petr. Kompromitace dat pomocí SQL Injection, část I. DSM Data Security Management. 2014, roč. 18, č. 1, s. 36-39. ISSN 1211-8737.

ANTAL Lukáš, BARABAS Maroš a HANÁČEK Petr. Kompromitace dat pomocí SQL Injection, část II. DSM Data Security Management. 2014, roč. 18, č. 2, s. 32-35. ISSN 1211-8737.

ANTAL Lukáš, BARABAS Maroš a HANÁČEK Petr. Kompromitace dat pomocí SQL Injection, část III. DSM Data Security Management. 2014, roč. 18, č. 3, s. 25-29. ISSN 1211-8737.

HENZL Martin, BARABAS Maroš, JANČA Radim a HANÁČEK Petr. Bezpečnost bezkontaktních platebních karet. DSM Data Security Management. 2014, roč. 28, č. 2, s. 41-43. ISSN 1211-8737.

KAČIC Matej, OVŠONKA Daniel, BARABAS Maroš a HANÁČEK Petr. Traffic generator based on behavioral pattern. In: Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for. London: IEEE Computer Society, 2014, s. 230-234. ISBN 978-1-908320-40-7.

DRAHANSKÝ Martin, HANÁČEK Petr, ZBOŘIL František V., ZBOŘIL František, BARABAS Maroš a ANTAL Lukáš. Threats in Networks using Agent and Biometric Systems. International Journal of Bio-Science and Bio-Technology. Daedok: Středisko pro podporu vědy a technického výzkumu, 2013, roč. 5, č. 3, s. 119-133. ISSN 2233-7849.

JURNEČKA Peter, HANÁČEK Petr, BARABAS Maroš, HENZL Martin a KAČIC Matej. A method for parallel software refactoring for safety standards compliance. In: System Safety 2013 collection of papers. Cardiff: The Institution of Engineering and Technology, 2013, s. 1-6. ISBN 978-1-84919-777-9. ISSN 0537-9989.

JURNEČKA Peter, HANÁČEK Petr, BARABAS Maroš, HENZL Martin a KAČIC Matej. A method for parallel software refactoring for safety standards compliance. Resilience, Security & Risk in Transport. London: The Institution of Engineering and Technology, 2013, s. 42-48. ISBN 978-1-84919-787-8.

BARABAS Maroš. Automated Processes in Computer Security. In: Proceedings of the 16th Conference STUDENT EEICT 2010. Brno: Vysoké učení technické v Brně, 2010, s. 246-250. ISBN 978-80-214-4079-1.

A.3 Ostatné publikačné aktivity

PORUBČAN Juraj a BARABAS Maroš. Ochrana citlivých informácií pred únikom z firmy. Infoware. Bratislava, 2014, č. 10/2014, s. 54. ISSN 1336-3581

PORUBČAN Juraj a BARABAS Maroš. Jak vyvíjet bezpečnější aplikace pomocí SSDLC (Secure Software Development Life Cycle). IT System. Brno, 2014, roč. 2014, č. 4, s. 32-33. ISSN 1212-4567

BARABAS Maroš a DROZD Michal. Pokročilé formy útoků a jejich detekce. IT System. Brno: CCB spol. s r.o., 2013, roč. 2013, č. 1, s. 8-10. ISSN 1212-4567.

BARABAS Maroš a DROZD Michal. Jak na pokročilé útoky. SecurityWorld. Praha: 2012, roč. 2012, č. 4, s. 20-20. ISSN 1802-4505.

A.4 Citácie

VAN LENTHE, J. M. Combining Multiple Malware Detection Approaches for Achieving Higher Accuracy. 2014.

CHAIRETAKIS, Eleftherios; ALKUDHIR, Bassam; MYSTRIDIS, Panagiotis. Deployment of Low Interaction Honeypots in University Campus Network. 2013.

HOMOLIAK Ivan, OVSONKA Daniel and GREGR Matej and HANACEK Petr. NBA of obfuscated network vulnerabilities' exploitation hidden into HTTPS traffic. In Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for, pp. 310-317. IEEE, 2014.

HOMOLIAK Ivan, OVSONKA Daniel, KORANDA Karel and HANACEK Petr. Characteristics of buffer overflow attacks tunneled in HTTP traffic. In Security Technology (ICCST), 2014 International Carnahan Conference on (pp. 1-6). IEEE.

TEKNOS Martin. Extension of Behavioral Analysis of Network Traffic Focusing on Attack Detection. In Excel Conference, Faculty of Information Technologies, Brno University Technology, 2012.

Dodatok B

Zoznam zraniteľných programov

Arkeia 5.3.3 and 5.2.27 Windows (All)
3Com 3CDaemon FTP service 2r10
BolinTech Dream FTP Server version 1.02
EasyFTP Server 1.7.0.11
freeFTPd multi-protocol file transfer service
Golden FTP Service
Sami FTP Server version 2.02
SlimFTPd Server 3.16 Universal
War-FTPD 1.65
Alt-N SecurityGateway 1.0.1 Universal
BadBlue 2.5 (Universal)
Windows Apache 2.2 version Universal
EFS Software Easy Chat Server 2.2
WhatsUP Gold 8.03 Universal
Minishare 1.4.1
PSO Proxy v0.91
Savant 3.1 Web Server
SHTTPD 1.34
Firebird WI-V2.0.0.12748 WI-V2.0.1.12855
Netcat v1.10 NT
Wireshark 1.4.4
MySQL 5.5.9
WinProxy 6.1 R1a Universal
WinGate 6.1.1.1077
Mercury Mail Transport System 4.51
FreeSSHd 1.0.9
FutureSoft TFTP Server 2000 1.0.0.1
TFTPDWIN v0.4.2
WinVNC Web Server v3.3.3r7
Kerio Personal Firewall 2.1.4
CesarFTP 0.99g
Easy File Sharing 2.0 service
fileCOPA FTP server pre 18 Jul 2006 version
GlobalSCAPE Secure FTP Server 3.0.2
HTTPDX FTP server 1.5
Serv-U 4.1.0.3
vftpd 1.31
WFTPD 3.23
mod_jk 1.2.20 (Apache 2.0.58)
BadBlue EE 2.7 Universal
EasyFTP Server 1.7.0.11
httpdx 1.4
Manage Engine Appplications Manager
PeerCast 0.1216
Sambar 6
SHOUTcast DNAS 1.9.4
Xitami 2.5c2 Web Server
Firebird WI-V1.5.3.4870 WI-V1.5.4.4910
Windows RSH daemon 1.8
Motorola's Timbuktu Pro for Windows 8.6.5
MySQL 6.0
CCProxy v6.2
sipXezPhone 0.35a Universal
YPOPS POP3 service v0.6
Allied Telesyn TFTP Server 1.9
TFTPD32 2.21
3CTftpSvc 2.0.1