
**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**
**BRNO UNIVERSITY OF TECHNOLOGY**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**
**FACULTY OF INFORMATION TECHNOLOGY**

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**
**DEPARTMENT OF COMPUTER SYSTEMS**


# ODVOZOVÁNÍ PRAVIDEL PRO MITIGACI DDOS

**DERIVING DDOS MITIGATION RULES**

**DIPLOMOVÁ PRÁCE**
**MASTER'S THESIS**


**AUTOR PRÁCE** — **Bc. MAREK HURTA**
**AUTHOR**

**VEDOUCÍ PRÁCE** — **Ing. MARTIN ŽÁDNÍK, Ph.D.**
**SUPERVISOR**



**BRNO 2017**

*Zadání diplomové práce/19930/2016/xhurta01*

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačových systémů Akademický rok 2016/2017

# Zadání diplomové práce

Řešitel: **Hurta Marek, Bc.**
Obor: Bezpečnost informačních technologií
Téma: **Odvozování pravidel pro mitigaci DDoS**
**Deriving DDoS Mitigation Rules**
Kategorie: Počítačové sítě

Pokyny:

1. Nastudujte problematiku měření síťového provozu pomocí technologie NetFlow. Seznamte se s nejčastějšími síťovými bezpečnostními událostmi, které lze v NetFlow datech detekovat.
2. Seznamte se se systémem zpětného záchytu dat pomocí tzv. Time machine.
3. Navrhněte metodu pro odvozování mitigačních pravidel ze zachyceného provozu. Cílem je generovat taková pravidla, která pokryjí pouze nežádoucí provoz a zároveň počet těchto pravidel bude v řádu stovek.
4. Navržené řešení implementujte.
5. Funkčnost implementace demonstrujte na vhodně zvoleném vzorku dat.
6. Zhodnoťte dosažené výsledky a navrhněte možná rozšíření.

Literatura:

- Dle pokynů vedoucího práce.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

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

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

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

Vedoucí: **Žádník Martin, Ing., Ph.D.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 24. května 2017

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

L.S.

prof. Ing. Lukáš Sekanina, Ph.D.
*vedoucí ústavu*

# Abstrakt

Táto práca sa zaoberá monitorovaním sietí pomocou NetFlow dát. Popisuje princípy, na ktorých je založená detekcia bezpečnostných anomálií pomocou IDS systémov. Ďalej popisuje framework Nemea, ktorý slúži na tvorbu modulov schopných detekovat bezpečnostné anomálie na sieti. Následne sa venuje prehľadu jednotlivých útokov kde objasňuje ich špecifické vlastnosti, ako aj možné postupy pri ich analýze. Na základe tejto analýzy je možné vytvoriť sadu mitigačných pravidiel, ktorých aplikáciou môže dôjsť k zmierneniu prebiehajúceho útoku. Na základe získaných poznatkov bol vytvorený návrh systému, ktorý bude schopný vytvárať mitigačné pravidlá automaticky. Pomocou navrhnutej metódy boli vykonané experimenty, pri ktorých metóda označila očakávané množstvo podozrivých dát.



# Abstract

This thesis is aimed at monitoring of computer networks using NetFlow data. It describes main aspects of detection network anomalies using IDS systems. Next part describes Nemea framework, which is used for creating modules. These modules are able to detect network incidents and attacks. Following chapters contain a brief overview of common network attacks with their specific remarks which can help in process of their detection. Based on this analysis, the concept of mitigation rules was created. These rules can be used for mitigation of DDoS attack. This method was tested on several data sets and it produced multiple mitigation rules. These rules were applied on data sets and they marked most of the suspicious flows.



# Kľúčové slová

NetFlow, IDS systémy, DDoS útok, Systém spätného záchytu, Mitigačné pravidlá

# Keywords

NetFlow, IDS systems, DDoS attack, Time machine system, Mitigation rules


# Citácia

HURTA, Marek. *Odvozování pravidel pro mitigaci DDoS*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Žádník Martin.

# Odvozování pravidel pro mitigaci DDoS

## Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Martina Žádníka, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Marek Hurta
17. mája 2017

# Obsah

# Kapitola 1

# Úvod

V dnešnej dobe sa čoraz viac stretávame s bezpečnostnými incidentami, ktorých cieľom je čiastočne alebo úplne obmedziť používanie rôznych druhov služieb, či napríklad obmedziť možnosť prístupu k webových serverom. Tento typ incidentov môžeme označiť ako DDoS (*angl. Distributed Denial of Service*) útok. Základným princípom je preťaženie cieľovej stanice, na ktorej je služba prevádzkovaná napríklad vyčerpaním jej HW zdrojov, čo v konečnom dôsledku vedie k neschopnosti odpovedať na požiadavky bežných užívateľov.

Z tohto dôvodu je kladené čoraz väčšie úsilie na tvorbu systémov, ktoré by boli schopné tieto druhy útokov detekovať a prípadne proti nim adekvátne zasiahnuť. Tieto systémy sa nazývajú IDS (*angl. Intrusion Detection System*). Jedným z nich je aj modulárny systém NEMEA (*Network Measurements Analysis*), ktorý zároveň tvorí framework pre tvorbu nových detekčných modulov slúžiacich na odhaľovanie rôznych druhov útokov z NetFlow dát. V kombinácií so systémom spätného záchytu dát *Time Machine*, ktorý dokáže zachytiť dátový tok prebiehajúceho útoku je takýto systém schopný obmedziť prebiehajúci útok.

Na to aby bol tento systém schopný efektívne fungovať je potrebné vytváranie mitigačných pravidiel, ktoré určitým spôsobom definujú aký typ komunikácie je potrebné obmedziť za účelom zmiernenia účinkov prebiehajúceho DDoS útoku. Návrhom týchto pravidiel ako aj tvorbou samotného generátoru mitigačných pravidiel sa zaoberá práve táto práca.

Kapitola 2 sa zaoberá základnými prístupmi pri monitorovaní sieťovej komunikácie. Popisuje základné časti protokolu NetFlow ako aj jeho architektúru a niektoré jeho verzie. Záver kapitoly stručne popisuje protokoly IPFIX a sFlow.

V kapitole 3 budú predstavené princípy detekcie bezpečnostných hrozieb na sieti. Čitateľ bude oboznámený s princípom činnosti IDS systémov ako aj z ich základným rozdelením. Obsahom kapitoly je tiež predstavenie dvoch najznámejších zástupcov týchto systémov spolu s popisom ich činnosti. Záver kapitoly sa venuje frameworku NEMEA. Bude vysvetlený princíp komunikácie medzi jednotlivými časťami a predstavený formát UniRec. Taktiež bude podrobne popísaný systém spätného záchytu dát *Time Machine*. Kapitola 4 obsahuje popis niektorých nástrojov používaných pri analýze dát. Ďalej budú predstavené vybrané druhy bezpečnostných anomálií spolu s ich základnými charakteristikami.

V kapitole 5 bude objasnený princíp na základe ktorého sú vytvárané mitigačné pravidlá a taktiež budú popísané jednotlivé položky, z ktorých sú pravidlá zložené. V naväzujúcej kapitole bude prezentovaný postup implementácie jednotlivých častí navrhnutej metódy. V závere kapitoly je v krátkosti načrtnutý princíp, ktorým sa overuje efektivita vybraných položiek mitigačného pravidla.

Záverečná kapitola prezentuje metodiku, podľa ktorej boli vykonané experimenty za účelom demonštrácie funkcionality navrhnutej metódy. Takisto budú prezentované dosiahnuté

výsledky, pre zvolené testovacie datové sady. Záver kapitoly pratrí krátkemu porovnaniu navrhnutej metódy s alternatívnymi prístupmi pri analýze dát a stručnému zhodnoteniu dosiahnutých výsledkov.

Táto práca vychádzala zo zadania Semestrálneho projektu, ktorý sa zaoberal popisom použitých technológií a čiastočným návrhom tvorby mitigačných pravidiel.

# Kapitola 2

# Metódy monitorovania siete

V dnešnej dobe sa pomocou počítačových sietí prenáša čoraz väčšie množstvo dát a s tým je priamo spojená aj potreba získavania informácií, ktoré slúžia pri monitorovaní a správe týchto sietí. Takisto je možné pomocou získaných informácií vykonávať rôzne administratívne úkony. Nakoľko objem týchto informácií z roka na rok rastie, rastie aj potreba spracovávať tieto informácie automatizovaným spôsobom v ideálnom prípade s čo najmenšími zásahmi človeka.

V tejto kapitole budú naznačené základné prístupy pri monitorovaní siete. Taktiež bude predstavený protokol NetFlow, jeho základná štruktúra spolu s jeho verziami. V závere kapitoly sú popísané ďalšie protokoly, s ktorými sa môžeme stretnúť pri monitorovaní sietí a to IPFIX a sFlow.

## 2.1 Rozdelenie monitorovania siete

V rámci monitorovania siete existujú dva základné prístupy[14]:

- **Pasívne monitorovanie** - Princípom tohto typu monitorovania je zbieranie dostatočného množstva informácií o prebiehajúcej komunikácií na sieti. Informácie môžu pochádzať z logovacích záznamov služieb, či aplikácií. Môžu to byť asynchronné SNMP správy tzv. *traps*, prípadne NetFlow informácie, ktoré obsahujú štatistiky o danej komunikácií. Z hľadiska ďalšieho spracovania sú NetFlow dáta výhodné hlavne pri analýze sieťovej komunikácie, napríklad za účelom detekcie sieťových anomálií.
- **Aktívne monitorovanie** - Na rozdiel od pasívneho typu monitorovania pri tomto type dochádza ku aktívnemu prístupu zo strany administrátora, ktorý sa snaží zisťovať aktuálny stav siete a jej prvkov. K tomu sa najčastejšie využivajú protokoly ICMP a Telnet. Zisťovanie dostupnosti prebieha periodicky a jeho cieľom je čo najrýchlejšie odhaliť výpadok služby, aplikácie alebo sieťového prvku a informovať o tomto výpadku formou správy. Taktiež je administrátor schopný získať rôzne informácie o stave a konfigurácii jednotlivých prvkov.

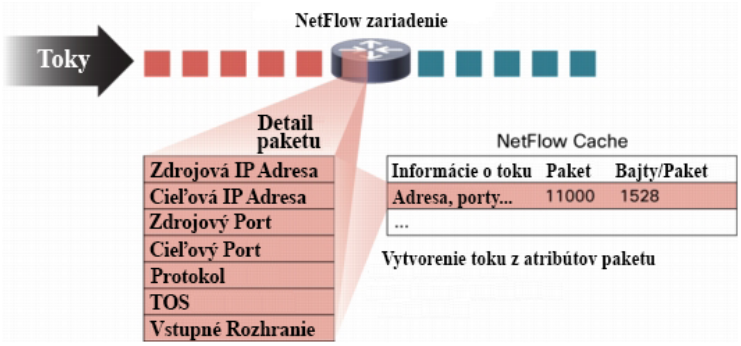
## 2.2 NetFlow

NetFlow je sieťový protokol vyvinutý firmou Cisco, ktorý slúži na zbieranie štatistík o tokoch v počítačovej sieti. Princípom tejto technológie je zbieranie len vybraných informácii o tokoch, čím sa redukuje objem ukladaných dát a je možné túto technológiu použiť hlavne vo

vysokorýchlostných sieťach, kde by zachytávanie detailných informácii nebolo z pamäťového i výpočtového výkonu vôbec možné.

Sieťový tok je definovaný ako jednosmerná postupnosť IP paketov prechádzajúcich určitým bodom v sieti za jednotku času[10]. Zároveň je možné každý NetFlow tok jednoznačne identifikovať na základe sedmice vlastností, ktoré je možné vidieť na obrázku 2.1 a sú nasledovné:

- Zdrojová IP adresa
- Cieľová IP adresa
- Zdrojový port
- Cieľový port
- Číslo protokolu (3. vrstva ISO/OSI modelu)
- Číslo vstupného rozhrania
- Typ služby (*angl. Type of service - ToS*)




Obr. 2.1: Štruktúra IP toku[3]

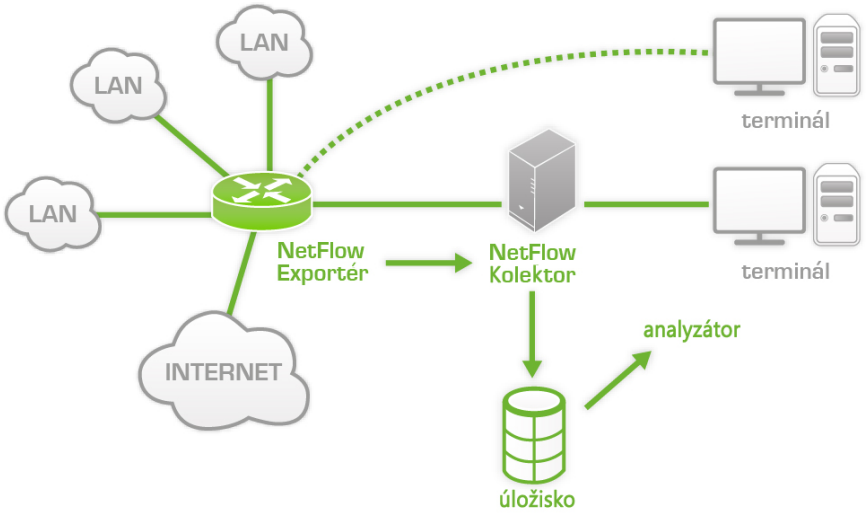
Tieto informácie slúžia predovšetkým na identifikovanie daného toku v rámci komunikácie medzi dvoma komunikujúcimi strojmi. V prípade hĺbkovej analýzy sieťovej komunikácie je potrebné uchovávať aj ďalšie doplňujúce štatistiky ako počet prenesených paketov, ich veľkosť, časové značky a iné. Tieto štatistiky v kombinácií s vyššie uvedenou sedmicou tvoria základ pre prípadnú analýzu komunikácie a sú ukladané do tzv. NetFlow Cache, ktorá sa nachádza v každom NetFlow zariadení.

## 2.3 Architektúra NetFlow

Základom celej architektúry je zariadenie, ktoré umožnuje zachytávanie samotných dát v určitom bode siete. Vo väčšine prípadov sa tieto zariadenia umiestňujú na hraničné body medzi dvoma sieťami, čím je zaistené monitorovanie komunikácie, ktorá do siete na tomto mieste vchádza a vychádza. Takisto je možné tieto zariadenia nasadiť aj v ľubovoľnom mieste v sieti.

Zachytené dáta sú pomocou protokolu NetFlow posielané na kolektor, kde sa ukladajú v závislosti od zvolenej technológie do súborov na disk prípadne do databázy. Architektúra taktiež podporuje rôzne druhy zapojenia, kedy je možné zasielanie NetFlow dát z viacerých

exportérov na jeden alebo viacero kolektorov. Jednoduchý typ prepojenia je možné vidieť na obrázku 2.2.




Obr. 2.2: Architektúra NetFlow[2]

### 2.3.1 Exportér

Exportér je sieťový prvok, prípadne SW zariadenie, ktoré umožňuje získavať informácie o linke na základe monitorovania paketov, ktoré skladá do spoločných tokov kde každý unikátny paket vytvára nový tok[10]. Jednotlivé toky sa ukladajú do NetFlow cache vo formáte ilustrovanom na obrázku 2.3. Spolu so základnými informáciami sú v cache uložené aj štatistiky pre daný tok.

Nakoľko je monitorovanie siete náročné z pohľadu veľkosti uchovávaných dát, je potrebné určitým spôsobom regulovať množstvo NetFlow záznamov v cache pamäti. Využíva sa preto princíp založený na exprirácii uložených štatistík, kedy v prípade, že niektorý zo záznamov spĺňa jednu z nižšie uvedených podmienok je odoslaný na kolektor a následne zmazaný z cache pamäte. Podmienky, pri ktorých dochádza k expirácii záznamu[10]:

- V prípade, že bol tok ukončený a prijatý paket obsahoval TCP príznak FIN označujúci ukončenie spojenia alebo RST označujúci reset spojenia[16].
- V prípade, že bola prekročená hodnota aktívneho časovača. Aktívný časovač reprezentuje dobu, počas ktorej môže byť záznam uložený v cache pamäti. Bežná doba expirácie je približne 30 minút, po ktorých uplinutí je záznam uvoľnený z pamäti. V prípade, že dorazí ďalší paket z toho istého toku, tento paket zakladá nový záznam v NetFlow cache pre daný tok.
- V prípade, že bola prekročená hodnota pasívneho časovača. Pasívný časovač určuje maximálnu dobu, počas ktorej môže byť záznam v cache pamäti neaktualizovaný. Počas tejto doby nieje prijatý ani jeden paket patriaci tomuto toku. Bežná hodnota časovača je 15 sekúnd.
- V prípade zaplnenia cache pamäti.

| SrcIf | SrcIPadd | DstIf | DstIPadd | Proto | ToS | Flgs | Pkts | Src Port | Src Msk | Src AS | Dst Port | Dst Msk | Dst AS | NextHop | Bytes/ Pkt | Active |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fa1/0 | 173.100.21.2 | Fa0/0 | 10.0.227.12 | 11 | 80 | 10 | 11000 | 162 | /24 | 5 | 163 | /24 | 15 | 10.0.23.2 | 1528 | 1745 |
| Fa1/0 | 173.100.3.2 | Fa0/0 | 10.0.227.12 | 6 | 40 | 0 | 2491 | 15 | /26 | 196 | 15 | /24 | 15 | 10.0.23.2 | 740 | 41,5 |
| Fa1/0 | 173.100.20.2 | Fa0/0 | 10.0.227.12 | 11 | 80 | 10 | 10000 | 161 | /24 | 180 | 10 | /24 | 15 | 10.0.23.2 | 1428 | 1145,5 |
| Fa1/0 | 173.100.6.2 | Fa0/0 | 10.0.227.12 | 6 | 40 | 0 | 2210 | 19 | /30 | 180 | 19 | /24 | 15 | 10.0.23.2 | 1040 | 24,5 |

Obr. 2.3: Položky uložené v NetFlow cache exportéru

### 2.3.2 Kolektor

NetFlow kolektor je zariadenie, ktoré umožňuje získavanie NetFlow záznamov z jedného alebo niekoľkých exportérov v závislosti na architektúre zapojenia[7]. Komunikácia medzi kolektorom a exportérom prebieha pomocou NetFlow protokolu, ktorý je popísaný v kapitole 2.4.

Komunikácia prebieha prostredníctvom transportného protokolu UDP na porte 2055. Bežne sa môžeme stretnúť aj s využitím iných portov. Tento protokol je však problematický z hľadiska spoľahlivosti a pokiaľ dôjde ku strate niektorých záznamov, kolektor nemá šancu opätovne získať tieto záznamy z exportéru, nakoľko každý odoslaný záznam je ihneď uvoľnený z NetFlow cache exportéru.

Preto je možné použiť protokol SCTP (Stream Control Transmission Protocol), ktorý garantuje bezstratové spojenie[20]. Z pohľadu bezpečnosti obsahuje tento protokol mechanizmy, ktoré dokážu čeliť niektorým bezpečnostným hrozbám, ako napríklad SYN Flood DoS útoku[6]. Nevýhodou však môže byť komunikácia všetkých zapojených kolektorov so všetkými NetFlow sieťovými prvkami, čím môže dochádzať k ich nadmernej záťaži a poklesu celkovej výkonnosti.

## 2.4 NetFlow protokol

Existuje niekoľko verzii tohto protokolu, pričom niektoré z nich neboli nikdy aplikované do reálneho prostredia a jednalo sa len o interné verzie firmy Cisco. Dnes patria medzi najrozšírenejšie hlavne verzie **v5** a **v9**. Verzia 9 tvorila základ pre vytvorenie nového protokolu IPFIX (Internet Protocol Flow Information Export), ktorý je štandardom IETF (Internet Engineering Task Force)[5]. Tomuto protokolu sa bližšie venuje kapitola 2.5.

### 2.4.1 Verzia 5

Táto verzia protokolu sa vyznačuje hlavne fixným formátom uloženia dát. Skladá sa z hlavičky, ktorá má veľkosť 24 bajtov a záznamu s veľkosťou 48 bajtov. Hlavnou nevýhodou je práve fixná veľkosť záznamu, čím je znemožnené pridávanie nových položiek[8].

Pokiaľ chceme zisťovať väčšie množstvo informácii o tokoch, narazíme pri tejto verzii práve na tento problém, ktorý bolo nutné vyriešiť návrhom novej, robusnejšej verzie protokolu, ktorá bude schopná pridávať jednotlivé položky dynamicky. Ďalšou nevýhodou je nepodporovanie IPv6 tokov. Pre ilustráciu môžeme uviesť položky, ktoré táto verzia obsahuje[5]:

- Zdrojová/Cieľová IP adresa, Zdrojový/Cieľový port, Číslo protokolu
- Zdrojová/Cieľová maska siete, Nasledujúci uzol (*angl. next hop*), Číslo autonómneho systému zdroja a cieľa

- Číslo vstupného a výstupného rozhrania
- ToS
- Počet bajtov a paketov v toku
- TCP príznaky
- Časové značky začiatku a konca toku

### 2.4.2 Verzia 9

Nová verzia protokolu oproti verzii 5 obsahuje flexibilný formát záznamu, ktorého položky je možné ľubovoľne meniť. Exportované záznamy sa opäť skladajú z hlavičky, ktorej veľkosť zostala rovnaká. Za hlavičkou sa nachádzajú šablóny nesledované dátami.

Každá šablóna definuje popis jednotlivých typov položiek, z ktorých sa skladá dátová časť[10]. Kolektor je na základe toho schopný spracovať dáta, ktoré sa nachádzajú v zázname. Pokiaľ kolektor narazí na položku, ktorú nepozná, vynechá ju a pokračuje v spracovávaní nasledujúcej položky.

Hlavnou výhodou je možnosť konfigurácie šablóny, ktorá umožní posielanie len tých položiek, ktoré su pre nás zaujímavé, čo v konečnom dôsledku môže viesť ku zredukovaniu objemu prenášaných dát. Okrem toho, bola do verzie 9 pridaná podpora IPv6 a MPLS (Multi Protocol Label Switching)[9].

## 2.5 IPFIX

Protokol IPFIX (IP Flow Information Export) vznikol ako následník protokolu NetFlow v9 a stal sa IETF štandardom. Hlavným prínosom bola možnosť definovať vlastné položky, ktoré nemusia odpovedať propietárnym položkám firmy Cisco. Tým pádom bolo umožnené používanie tohto protokolu aj na zariadeniach, ktoré nevyrába firma Cisco.

Pomocou tohto protokolu je teda možné exportovať všetky doterajšie položky zahrnuté v prokotoloch NetFlow a zároveň je možný export unikátnych položiek, ktoré sú špecifické pre zariadenia jednotlivých výrobcov. Okrem toho, IPFIX priniesol možnosť využívať pri komunikácii medzi exportérom a kolektorom prokoly TCP a SCTP (Stream Control Transmission Protocol)[11].

## 2.6 sFlow

Technológia sFlow bola navrhnutá pre monitorovanie tokov vo vysokorýchlostných sieťach[15]. Ide o technológiu, ktorá je odlišná od dvoch predchádzajúcich hlavne tým, že sa jedná o HW orientovné riešenie. Sieťový prvok, ktorý implementuje túto technológiu je zodpovedný za vytváranie a exportovanie sFlow datagramov. Tieto datagramy obsahujú časti flow záznamov v kombinácii s čítačmi rozhraní.

Technológia využíva na komunikáciu protokol UDP s číslom portu 634 a je navrhnutá tak, že prípadná strata paketu, ktorá je spôsobená nespoľahlivosťou transtportného protokolu neovplyvňuje celkovú funkčnosť. Architektúra je podobná ako u technológie NetFlow, v rámci sieťových prvkov tzv. Agentov prebieha záchyt, tvorba a exportovanie sFlow datagramov do sFlow Collectoru, kde sú tieto dáta ukladané a prebieha nad nimi analýza[19].

# Kapitola 3

# Princípy detekcie bezpečnostných hrozieb na sieti

Táto kapitola pojednáva o možnostiach akým spôsobom je možné čeliť bezpečnostným hrozbám, s ktorými sa dnes môžeme na sieti stretnúť. Základom je získanie relevantných dát, ktoré umožňujú vykonávanie detailnej analýzy za účelom odhalenia bezpečtnostných hrozieb. Zdrojom týchto dát je prevažne monitorovanie siete, ktoré bolo podrobne popísané v predchádzajúcej kapitole.

V tejto kapitole budú uvedené základné prístupy akými je možné odhaliť potenciálne hrozby. Podkapitola 3.1 sa zaoberá predsavením IDS systémov, ktoré slúžia na detekciu bezpečnostných horzieb. V ďalšej časti budú predstavené systémy Snort a Bro, ktoré patria medzi najznámejšie v tejto kategórii. V časti 3.4 bude predstavený framework NEMEA, ktorý umožňuje analýzu a detekciu bezpečnostných hrozieb s využitím NetFlow dát.

## 3.1 IDS systémy

IDS (Intrusion Detection System) je zariadenie alebo software, ktorý monitoruje počítačovú sieť a snaží sa nájsť potenciálne bezpečnostné hrozby v sieťovej komunikácii[13]. Tento systém môže byť použitý na detekciu podozrivej sieťovej komunikácie, útokov na sieťové služby ako napríklad skenovanie portov alebo štruktúry siete, útoky hrubou silou na bežne používané služby ako SSH, Telnet alebo RDP. Ďalej je schopný detekovať sieťové útoky ako napríklad DoS, DDoS, DNS amplifikáciu a mnoho ďalších.

Výstupom týchto systémov je väčšinou varovanie pre sieťových alebo systémových administrátorov, ktoré obsahuje relevantné informácie popisujúce typ hrozby, jej rozsah a mnoho ďalších informácii spojených s touto hrozbou. Na základe toho je administrátor schopný reagovať na vzniknutú situáciu podniknutím opatrení, ktoré by mali viesť k zamedzeniu tejto hrozby, prípadne k zmierneniu jej dopadov.

Oproti tomu, sa môžeme stretnúť so systémami IPS (Intrusion Prevention System), ktoré sa odlišujú od IDS systémov tým, že sú schopné priamo zasiahnuť do sieťovej komunikácie v prípade, že je táto komunikácia vyhodnotená ako nebezpečná. Inými slovami sú tieto systémy schopné zabolokovať sieťovú komunikáciu v prípade detekovania určitej hrozby. Existuje niekoľko pohľadov na rozdelenie IDS systémov, na základe ktorých vznikli dve kategórie, ktoré rozdeľujú tieto systémy podľa miesta ich nasadenia:

- **Host-based** - Ide o systém, ktorý je umiestnený priamo na užívateľských počítačoch alebo vybraných sieťových zariadeniach. Jeho úlohou je monitorovanie prichádzajúcich

a odchádzajúcich paketov zo zariadenia, vytváranie snímkov systému a konfiguračných informácii pre prípad nutnosti porovnania. Pokiaľ systém pri porovnávaní jednotlivých snímkov narazí na zmeny nastavení či konfiguračných súborov, ktoré sú vopred zadané ako nebezpečné, dôjde k vygenerovaniu upozornienia pre systémového administrátora, ktorému sú tieto zmeny poskytnuté na detailnejšie preskúmanie.

- **Network-based** - Tieto systémy sa nasadzujú na vybrané miesta v sieti, kde monitorujú prechádzajúcu komunikáciu a vyhodnocujú bezpečnostné udalosti, ktoré sú v prípade potreby schopné ohlasovať príslušnému administrátorovi. Výhodou tohto typu je možnosť nasadenia len na určitú časť siete, o ktorej monitorovanie sa zaujímame.

Ďalším kritériom pri rozdelení týchto systémov je spôsob akým pristupujú k monitorovaniu siete, ktoré im pomáha vytvárať celkový obraz o udalostiach na sieti a zároveň im umožňuje získať dostatok infromácii, potrebných k detekcii bezpečnostných hrozieb. Z hľadiska využitia detekčných metód, môžeme IDS systémy rozdeliť na dve hlavné skupiny:

- **Systémy založené na porovnávaní signatúr**
- **Systémy založené na vyhľadávaní anomálii**

### 3.1.1 Systémy založené na porovnávaní signatúr

Už z názvu týchto systémov (*angl. Signature-based*) vyplýva, že sú založené na princípe signatúr, ktoré určitým spôsobom charakterizujú podozrivú komiunikáciu. Signatúru si môžeme predstaviť ako súbor pravidiel, ktoré definujú hodnoty sledovaných vlastností komunikácie. Pokiaľ dôjde ku zhode aktuálne analyzovanej komunikácie s niektorou z uložených signatúr je táto komunikácia označená za podozrivú, prípadne nebezpečnú v závislosti od typu hrozby, ktorú predstavuje.

Úroveň porovnávania závisí od typu signatúr. Niektoré zo systémov využívajú signatúry popisujúce vlastnosti samotných paketov, iné sa môžu zameriavať na porovnávanie na základe celej sady pravidiel, ktorá obsahuje nielen signatúry pre jednotlivé pakety, ale aj signatúry popisujúce vlastnosti komunikácie v širšom slova zmysle. V praxi sa môžeme stretnúť s celými množinami signatúr, ktoré sú vytvárané práve za účelom jemnejšieho rozlýšenia bežnej a škodlivej komunikácie.

V tomto type systémov je obecne kladený veľký dôraz na detailnú špecifikáciu jednotlivých pravidiel, ktorými sa snažíme čo najviac vylepšiť danú signatúru a tým v konečnom dôsledku znížiť počet falošne pozitívnych detekcii.

Na nasledujúcom obrázku je možné vidieť príklad signatúry, ktorá popisuje SYN TCP sken z pohľadu IP tokov:

| Protokol | Pakety | Bajty | TCP príznaky | Toky |
|---|---|---|---|---|
| TCP | <1;2> | <40;80> | SYN | 1 |

Obr. 3.1: Príklad signatúry pre detekciu TCP SYN skenovania.

Hlavnou nevýhodou tejto metódy je neschopnosť detekovať nové druhy hrozieb, nakoľko sa v systéme nenachádza súbor signatúr popisujúcich nový typ hrozby. Z toho vyplýva aj fakt, že s každým novým typom útoku je potrebná analýza a návrh určitej množiny pravidiel, na základe ktorých bude možné vytvoriť signatúru popisujúcu tento útok.

Jedným z hlavných predstaviteľov tohto typu systémov je IDS systém **Snort**[1], ktorý popisuje kapitola 3.2.

### 3.1.2 Systémy založené na vyhľadávaní anomálii

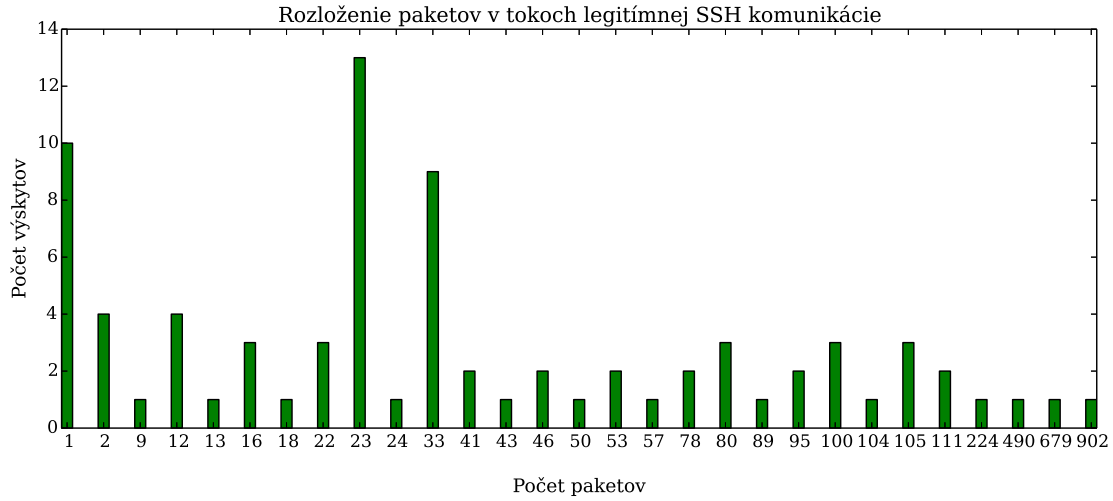
Tieto systémy (*angl. Anomaly-based*) sa líšia od predošlých hlavne tým, že detekcia hrozieb neprebieha na základe statických pravidel, ktoré je nutné dopredu zadať do detekčného systému ale zameriavajú sa hlavne na celkový obraz komunikácie. Následne sa snažia označiť tie zmeny v komunikácii, ktoré sa nejakým spôsobom vymykajú bežnému profilu komunikácie na sieti.[21]

Fungovanie týchto systémov je zvyčajne rozdelené do dvoch častí, kde v počiatočnej fáze si samotný systém vytvára profil sieťovej komunikácie, ktorý popisuje bežný stav na sieti bez bezpečnostných incidentov. Na základe toho je systém neskôr schopný porovnávať, či sa aktuálne analyzovaná komunikácia vymyká bežnému profilu. Táto analýza je zároveň druhou fázou.

Jednou z hlavých výhod je schopnosť reagovať na nové typy hrozieb, či priamo útokov bez zásahu človeka. Detekcia každej anomálie v komunikácii je považovaná za minimálne podozrivú. Z toho však vyplýva, že pokiaľ sa na sieti objaví určitý druh legitímnej komunikácie, ktorá sa však nejakým spôsobom vymyká štandartnému profilu komunikácie je pravdepodobné, že bude táto komunikácia označená za podozrivú.

S tým súvisí jedna z hlavných nevýhod týchto systémov a to je veľké množstvo falošne pozitívnych detekcii. Taktiež je problematická detekcia útokov, ktoré sa snažia svoje charakteristiky uričtým spôsobom maskovať tak, aby sa čo najviac priblížili profilu bežnej komunikácie.

Výstupom počiatočnej fázy nemusí byť komplexný obraz o celej sieťovej komunikácii ale môže sa jednať o profil, vytvorený z podmnožiny celkovej komunikácie. Tým pádom môžeme vytvoriť profil zachytávajúci bežnú komunikáciu len na špecifickom porte, prípadne protokole a na základe neho detekovať anomálie len na vybranej časti komunikácie. Takto vytvorený profil komunikácie na porte 22, ktorý bol vytvorený z NetFlow dát je možné vidieť na nasledujúcom obrázku:




Obr. 3.2: Profil legitímnej komunikácie na protokol SSH

[1] http://www.snort.org/

Pre IDS systémy založené na tomto princípe je veľmi dôležité mať prehľad o širokom spektre protokolov, služieb a ich rozličných nastaveniach. Táto podmienka by sa dala označiť za kľučovú, pokiaľ chceme aby nasadený systém pracoval čo najefektívnejšie.

V rámci sieťovej komunikácie dochádza veľmi často k vytváraniu neštandartných vzorov komunikácie, ktoré však nemusia znamenať žiadnu hrozbu. Pre príklad stačí uviesť spustenie služby určitého druhu na inom porte než je pre túto službu bežné. Tým pádom môže dôjsť k nárastu komunikácie na tomto porte a pre systém to môže indikovať potenciálnu hrozbu.

Preto je potrebné aby bol takýto systém nakonfigurovaný v rozumnej miere v závislosti na tom o aké druhy protokolov sa zaujímame a na aké druhy útokov sa prioritne zameriavame. Ľahko však môže nastať situácia, kedy počet generovaných hrozieb prekročí únosné množstvo a tým pádom administrátor nieje schopný reálne vyhodnotiť, ktorá z hrozieb je skutočne nebezpečná.

Jedným z najznámejších IDS systémov založených na tomto spôsobe detekcie je IDS systém **Bro**[2], ktorý bude popísaný v kapitole 3.3.

## 3.2 Snort

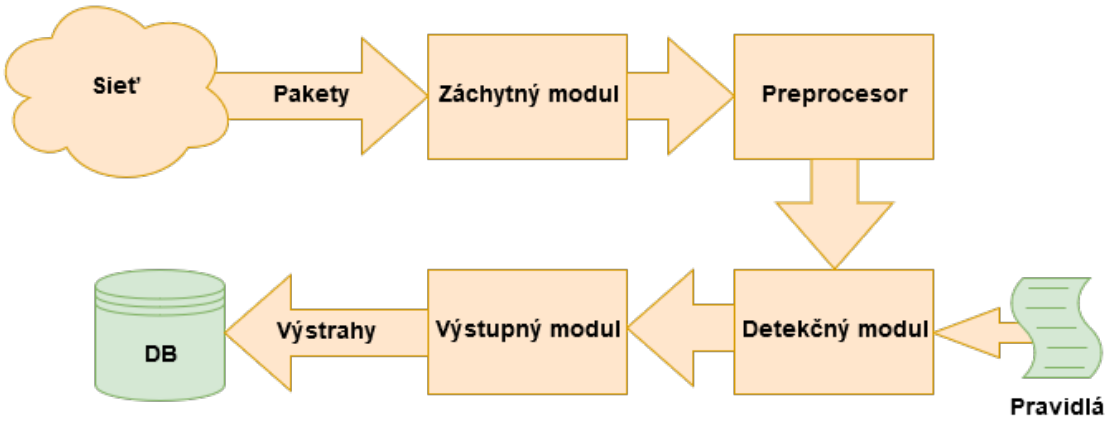
Snort sa radí do skupiny IDS systémov, ktoré pracujú na základe detekcie signatúr. Tento systém je založený na rýchlom zachytávaní paketov a ich porovnávaní oproti detekčným pravidlám. Systém dokáže v závislosti od detekčných pravidiel odhaliť rozličné druhy hrozieb od skenovania portov až po široké spektrum útokov ako napríklad DDoS, buffer owerflow a iné[17]. Architekúra systému je zobrazená na obrázku 3.4 a dá sa rozdeliť na 4 základné časti:

- **Záchytný modul -** Jeho úlohou je zachytávanie paketov zo sieťovej komunikácie. Tento modul môže byť takisto využitý na zachytávanie a ukladanie paketov pre offline analýzu.
- **Preprocesor -** Hlavnou úlohou tejto komponenty je úprava extrahovaných paketov do podoby, vhodnej na ďalšie spracovanie. Jednotlivé zásuvné moduly kontrolujú prichádzajúce pakety a tie sú prostrednícvom nich posielané ďalej do systému. Tento návrh umožnuje tvorbu vlastných komponentov a ich zapojenie do celého systému.
- **Detekčný modul -** Po prijatí paketu od preprocesora dochádza ku zahájeniu porovnávania dát s detekčnými pravidlami. Počet pravidiel sa môže líšiť v závislosti od typu hrozby. Pokiaľ dôjde ku zhode dát a detekčných pravidel je vygenerované upozorienie, ktoré je poslané do výstupného modulu. Formát jednoduchého pravidla je možné vidieť na obrázku 3.3
- **Výstupný modul -** Tento modul má na starosti hlavne generovanie informatívneho výstupu na základe výsledku porovnaní v detekčnom module. Výstupom modulu môže byť jednoduchý logovací súbor, obsahujúci relevantné informácie k detekovanej udalosti. Tak isto je možné vygenerovanú výstrahu poslať prostredníctvom klasického UNIX socketu na vzdialený počítač. Ďalej je možné ukladanie priamo do databázy alebo generovanie upozornení pomocou Syslogu.

[2] http://www.bro.org/

| Akcia | Protokol | Zdrojová IP | Zdrojový port | Smer | Cieľová IP | Cieľový port | Vlastnosti |
|---|---|---|---|---|---|---|---|
| drop | tcp | any | any | -> | 10.11.12.13 | 22 | (content:"/bin/sh"; msg:"Warning, possible SSH buffer overflow";) |

Obr. 3.3: Príklad detekčného pravidla.




Obr. 3.4: Architektúra systému Snort.

## 3.3 Bro

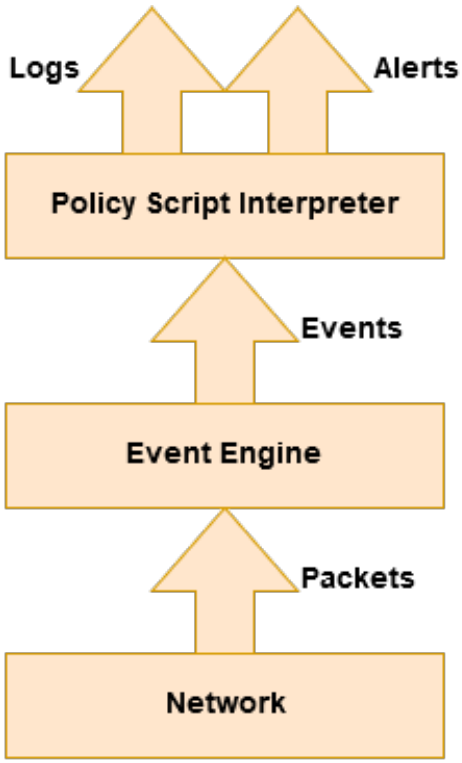
Bro je open source IDS systém, ktorý zachytáva sieťovú komunikáciu a na základe analýz nad týmito dátami sa snaží detekovať bezpečnostné hrozby v reálnom čase. Tento systém ponúka taktiež možnosť generovania štatistík o zachytenej komunikácii, ktoré môžu napríklad slúžiť aj ako dáta pre analýzu výkonnosti siete. Podporuje široké spektrum protokolov, služieb a vytvára formátovaný výstup vo forme logovacích súborov, ktoré môžu byť ďalej použité na strojové spracovanie.[1]

Návrh tohto systému umožňuje definovanie vlastných analýz, ktoré je možné zakomponovať do procesu detekcie. Tým pádom môžeme tento systém určitým spôsobom považovať za framework slúžiaci k detekovaniu bezpečnostných anomálii.[17] Architektúra systému je ilustrovaná na orázku 3.5 a zakladá sa na dvoch základných komponentoch:

- **Event Engine -** Má na starosti filtrovanie prichádzajúceho toku paketov zo siete na základe toho o akú detekciu sa zaujímame. Prichádzajúce pakety sú radené do skupín (*Events*) a tie sú následne preposielané ďalej do systému. Dôležitou súčasťou tohto komponentu, je knižnica LIBPCAP, ktorá umožňuje efektívne filtorvanie týchto paketov a taktiež tvorí jednotné rozhranie, ktoré je možné využiť v prípade, že chceme analyzovať už zachytenú komunikáciu vo forme PCAP súborov. Tento modul sa teda stará len o správne zatriedenie paketov do skupín, avšak nerieši ich sémantiku a takisto ani ich ďalší účel v procese spracovania.

- **Policy Script Interpreter -** Táto časť systému vykonáva nad prichádzajúcimi skupinami dát rôzne sady analýz, ktoré majú za účel zistiť povahu týchto dát. Jednotlivé analýzy sú popísané pomocou vlastného skriptovacieho jazyka, ktorý Bro ponúka. V rámci tohto modulu je teda možné vytvárať vlastné skripty, ktoré pracujú nad pri-

chádzajúcimi dátami. Taktiež je možné použiť širokú sadu predpripravených metód, ktoré sú už priamo zakomponované v systéme. Výstupom tohto modulu je záznam, ktorý v závislosti na nastavení obsahuje informácie spojené s výsledokm analýzy. Rovnako je možné v rámci výstupu generovať rôzne druhy upozornení pre ďalšie systémy a vďaka vlastnému skriptovaciemu jazyku je napríklad možné v prípade detekcie hrozby zároveň spúšťať niektoré externé programy alebo skripty.



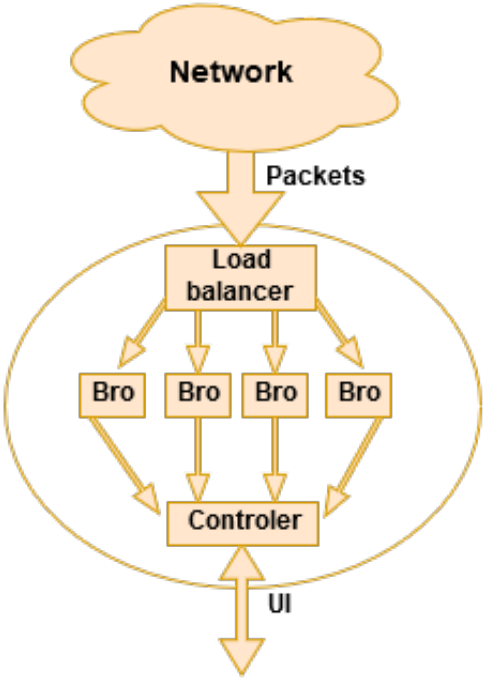

Obr. 3.5: Architektúra systému Bro.

Nasadenie tohoto systému je možné v rôznych druhoch sietí bez obmedzenia na ich veľkosť alebo zložitosť. V prípade veľmi veľkých sietí je možné nasadiť systém do tzv. *Clustru*. Toto riešenie sa od klasického nasadenia líši hlavne v tom, že na analýzu je využitých niekoľko Bro systémov súčastne.

Veľké množstvo paketov je distribuované na všetky aktívne podsystémy pomocou zariadenia nazývaného *load balancer*, ktorého hlavnou úlohou je rovnomerné distribuovanie paketov tak, aby nedochádzalo k zahlteniu niektorého z podsystémov. Následne sú pakety spracované a analyzované bežným spôsobom.

Výstupy z každého podsystému sú posielané do kontrolera, ktorý slúži ako zberný bod a zároveň je aj vstupným bodom pre užívateľské rozhranie. Užívateľ má prostredníctvom neho možnosť zobrazovať obsah logovacích súborov, prezerať detekované hrozby, atď.

Na obrázku 3.6, ktorý znázorňuje schému popisovaného zapojenia je možné vidieť jednotlivé komponenty a ich prepojenie. Takto nasadené riešenie je schopné komunikovať aj s ďalšími Bro systémami v prípade, že pre určitú časť siete chceme používať výhradne len jeden systém zameraný na detekciu hrozieb pre špeficický port alebo službu.




Obr. 3.6: Nasadenie systému Bro v Clustri.

## 3.4 NEMEA

Nemea (*Network Measturements Analysis*) je framework umožňujúci analýzu NetFlow dát získaných zo siete za účelom detekcie bezpečnostných hrozieb v reálnom čase. Architektúra tohto systému je navrhnutá s cieľom efektívneho vývoja nových druhov detekčných metód, ktoré sú reprezntované pomocou modulov. Jednotlivé moduly sú schopné komunikovať medzi sebou prostredníctvom zasielania správ.

### 3.4.1 Architektúra

Základnou jednotkou systému je samotný modul, ktorý je definovaný svojou funkcionalitou a rozhraním. Modul je samostatná jednotka, ktorá vykonáva v závislosti na implemnetácii rôzne druhy úloh.[4] Základné rozdelenie modulov je nasledujúce:

- **Detektor -** Vykonáva detekciu bezpečnostných hrozieb ako napríklad útoky DDoS, slovníkové útoky, DNS amplifikácia a iné.
- **Modul -** Predspracovanie vstupných dát (filtrovanie, agregácia, atď.), exportovanie a ukladanie výsledkov detekcii, tvorba reportov a logovacích súborov.

Ďalšou dôležitou časťou je samotný NEMEA framework, ktorý je taktiež možné rozdeliť na dve základné časti:

- **TRAP -** *Traffic Analysis Platform* tvorí základ celého frameworku, zodpovedá za prepojenie a komunikáciu jednotlivých modulov.
- **UniRec -** Implementuje efektívny formát správ, zasielaných medzi modulmi prostredníctvom TRAP rozhrania.

Medzi hlavné črty tohto frameworku patrí jeho modulárnosť, ktorá je zabezpečená práve jednotným rozhraním, ktoré implemnetujú jednotlivé moduly. Tým pádom je možné ľubovoľne kombinovať zapojenie modulov, či už za účelom detekcie alebo analýzy. Jedinou podmienkou, ktorá musí byť dodržaná na to aby bolo možné využiť niekoľko modulov súčastne je fakt, že formát výstupných a vstupných dát pri dvoch prepojených moduloch musí byť totožný. Takýmto spôsobom je možné prepojiť teoreticky neobmedzené množstvo modulov.

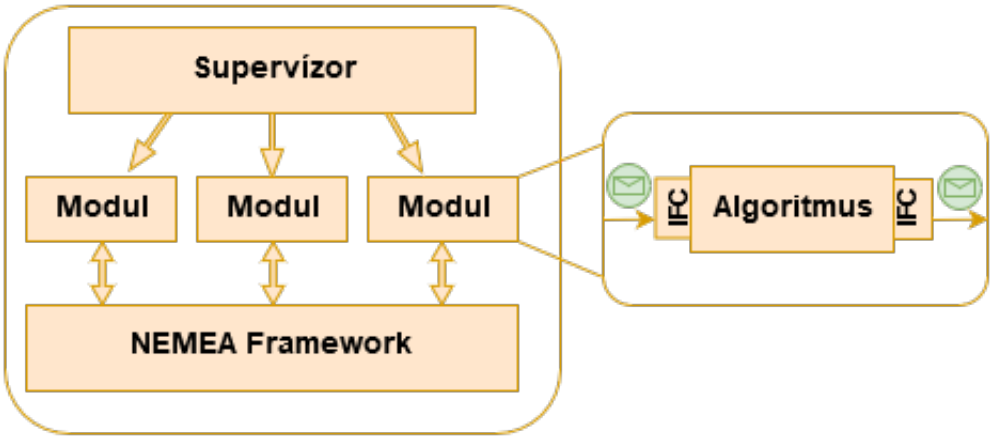
Modularita systému je výhodná z pohľadu samotnej detekcie, nakoľko je napríklad možné kedykoľvek pridať nový modul obsahujúci novú detekčnú metódu prípadne iné vylepšenie bez nutnosti ukončenia už prebiehajúcej analýzy. Medzi ďalšie výhody tohto návrhu patrí určite možnosť programovania modulov v rozličných programovacích jazykoch ako C, C++ alebo Python.

Samotné nasadenie systému je možné realizovať dvomi spôsobmi. Prvým spôsobom je manuálne spustenie modulu. Tento spôsob sa využíva v prípadoch kedy je detekčná metóda obsiahnutá v rámci jedného modulu, prípadne na jej realizáciu nieje potrebný žiadny ďalší modul. Tento prístup je takisto možné použiť aj v prípade vývoja modulov na účely rýchleho testovania.

Druhým spôsobom je spúšťanie prostredníctvom tzv. *Supervízora*. Jedná sa o program, ktorý je zodpovedný za konfiguráciu, spúšťanie, monitorovanie a údržbu väčšieho množstva modulov spustených súčastne. Tento druh zapojenia ilustruje obrázok 3.7.

Niektoré detekčné metódy vyžadujú zapojenie väčšieho množstva modulov súčastne preto by ich manuálna konfigurácia a hlavne údržba bola značne komplikovaná. Pomocou konfiguračného súboru je možné nastaviť jednotlivé parametre každého modulu.

Takisto sú počas detekcie zaznamenávané štatistiky o behu modulov a v prípade, že dôjde k zastaveniu niektorého z nich, je supervízor schopný tento modul okamžite reštartovať bez nutnosti prerušenia ostatných modulov.




Obr. 3.7: Zapojenie modulov pomocou Supervízora.[4]

### 3.4.2 Komunikácia medzi modulmi

Knižnica TRAP implementuje hlavné fukncie spojené s prepojením a komunikáciou medzi jednotlivými modulmi. Jedná sa o obojsmerné rozhranie, ktoré reprezentuje vstup a výstup každého modulu. Na obrázku 3.7 je možné toto rozhranie vidieť pod skratkou IFC (*TRAP Communication Interface*).

Moduly si môžu medzi sebou vymienať správy obsahujúce nielen samotné Netflow dáta, ale aj mnoho ďalších informácii ako napríklad výsledky detekcii, štatistické informácie získané z prichádzajúcich dát, detegované hrozby a iné. Posielané správy sú väčšinou vo formáte UniRec, ktorý bude popísaný v nasledovnej kapitole.

Ďalší podporovaný formát je JSON a rovnako je tiež možné využiť aj posielanie "surových"dát. IFC je navrhnuté tak, že vytvára abstrakciu nad procesom pripájania a komunikácie medzi modulmi a tým pádom značne uľahčuje prácu na vývoji modulu, nakoľko nieje potrebné riešiť tieto úlohy vlastnoručne.

Pri spúštaní modulu je však potrebné presne definovať rozhranie, z ktorého bude modul príjmať vstupné dáta a rozhrania kde bude posielať výstupné dáta. V rámci knižnice TRAP sú využité dva základné sockety, zabezpečujúce komunikáciu:

- **TCP socket -** Využíva sa pri komunikácii modulov prostredníctvom siete v prípade, že sú moduly spustené na rozdielnych strojoch.
- **UNIX socket -** Využíva sa pri komunikácii modulov v rámci toho istého fyzického stroja.

Okrem toho je možné využiť aj dve špeciálne IFC rozhrania, typ *file* a *blackhole*. Ako je už zo samotného názvu jasné, prvý typ reprezentuje klasické ukladanie výstupu do súboru. Tento súbor môže byť neskôr použitý na ďalšie spracovanie. Druhý typ je možné použiť len ako výtup, nakoľko sa správa podobne ako klasický linuxový **/dev/null** súbor.

### 3.4.3 Formát UniRec

Formát UniRec (*Unified Record*) je jedným z troch formátov správ posielaných prostredníctvom TRAP rozhrania. Tento formát sa skladá z položiek (*fields*), ktoré obsahujú posielané dáta. Aby bolo možné efektívne pristupovanie k jednotlivým položkám, je formát správy popísaný pomocou šablóny (*template*).

Princíp, na ktorom je tento formát založený približne zodpovedá formátu štruktúry v programovacom jazyku C. To umožňuje použiť priamy prístup k položkám, čo výrazne zefektívňuje celý proces. V porovnaní s ostatnými formátmi odpadá nutnosť spracovania správy pomocou parseru.

Oproti klasickej štruktúre je však v tomto formáte možné definovať aj položky dynamickej veľkosti. Medzi ďalšie výhody patrí napríklad možnosť vytvárania celej šablóny za behu programu. Z toho ale vyplýva, že každé IFC rozhranie dokáže odosielať len také správy, ktoré zieľajú spoločnú šablónu.

Pri nadviazaní spojenia medzi modulmi prebieha kontrola na úrovni IFC rozhrania, kde každé výstupné rozhranie špecifikuje formát dát, ktoré je schopné zasielať. Každé vstupné zariadenie musí špeficikovať aké položky očakáva na vstupe. Pri vytvorení spojenia dochádza ku kontrole týchto položiek a pokiaľ sú všetky požadované položky výstupného rozhrania obsiahnuté vo vstupnom formáte rozhrania, je možné nadviazať spojenie.

### 3.4.4 Systém spätného záchytu

Ako už bolo uvedené, detegovanie bezpečnostných hrozieb je v systéme NEMEA založené na dátach vo formáte NetFlow. Tento formát ponúka mnoho šatistík, ktoré sa dajú využiť na odhalenie rôznych druhov útokov.

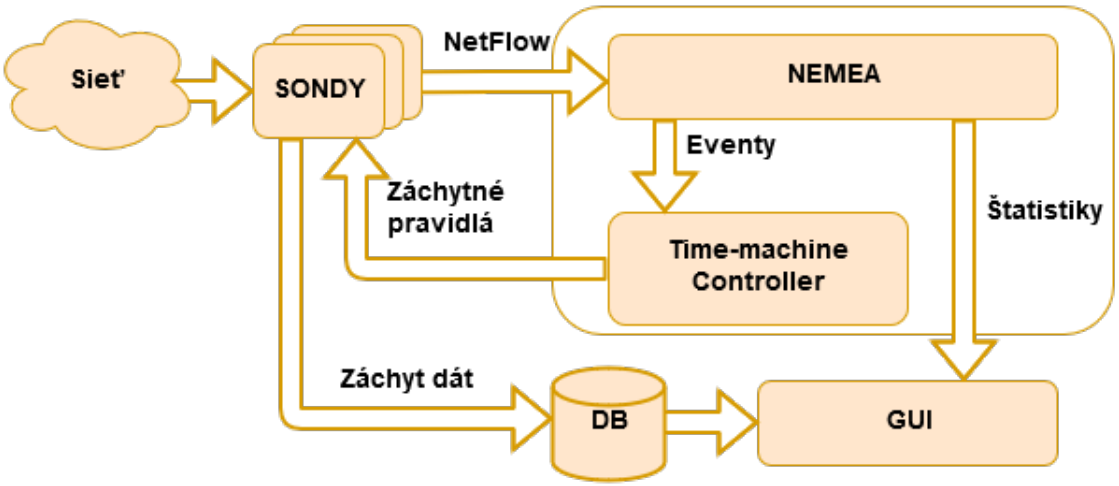
Niekedy však môžeme naraziť na problém, ktorým je absencia detailnejšich informácii o analyzovanej komunikácii. Tieto informácie by nám v niektorých prípadoch mohli výrazne

uľahčiť rozhodovanie, či je analyzovaná komunikácia naozaj nebezpečná a jedná sa o nejaký typ útoku alebo ide o bežnú komunikáciu, ktorá sa však svojou povahou podobá útoku. Tieto informácie však nieje často možné overiť, pokiaľ nemáme prístup ku kompletnej zachytenej vzorke dát.

Aj z toho dôvodu, bola vyvinutá technológia s názvom *Time machine*, ktorá umožňuje v určitých prípadoch zachytiť kompletný dátový tok prebiehajúcej komunikácie a tým poskytnúť dostatočné množstvo relevantných dát k následnej analýze.

Celý proces začína generovaním výstrahy z niektorého z detekčných modulov NEMEA. Táto správa väčšinou obsahuje typ útoku, IP adresu podozrivého stroja, prípadne IP adresu obete a ďalšie doplňujúce štatistiky v závislosti od typu hrozby. Správa je následne doručená do bloku s názvom *Time machine Controller*, ktorý je možné vidieť na obrázku 3.8.

Táto komponenta má za úlohu transformovať správu do formátu záchytných pravidiel, ktoré sú odoslané prostredníctvom zabezpečeného kanálu do sondy, ktorá disponuje systémom spätného záchytu.[18] Architektúru riešenia je možné vidieť na nasledujúcom obrázku:



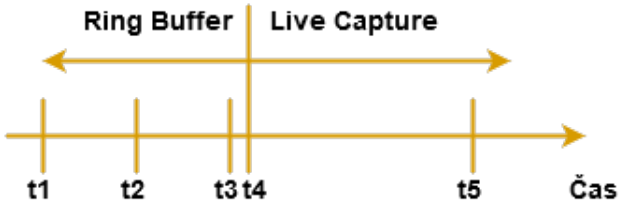

Obr. 3.8: Architektúra riešenia s využitím systému spätného záchytu.

Podstatnou časťou riešenia je upravená sonda založená na koncepte SDM (*Software Defined Monitoring*). Táto sonda okrem iného obsahuje aj jednotku záchytu paketov, ktorá je kľúčovou časťou riešenia. Jednotka záchytu je tvorená dvomi základnými komponentami:

- **Ring Buffer -** Ide o kruhový buffer, ktorý slúži na záchyt prvých N paketov toku na základe záchytného pravidla. Pakety sa ukladajú do RAM pamäte.
- **Live Capture -** Jednotka zachytávajúca všetky pakety komunikácie na základe záchytného pravidla. Pakety sú ukladané do súboru na disk.

Princíp zachytenia paketov na základe záchytných pravidiel v časovom intervale ilustruje obrázok 3.9. Bod v čase **t1** predstavuje začiatok bezpečnostnej hrozby, ktorej trvanie končí v čase **t5**. Po uplynutí času v bode **t2** sú pakety exportované zo sondy vo forme NetFlow záznamov na kolektor. V čase **t3** dochádza k detekcii samotnej hrozby. Z toho je možné odvodiť čas potrebný na detekciu, ktorý činí rozdiel časov **t3 - t1**. Je však možné vidieť, že samotný čas detekcie môže byť oveľa menší, nakoľko je nutné zohľadniť aj čas potrebný na exportovanie záznamov, ktorý je vyjadrený časovým rozdielom bodov **t2 - t1**.

V bode **t4** dochádza ku začiatku záchytu všetkých peketov, prisluchajúcich k danej hrozbe. Keďže *Ring Buffer* ukladá začiatočné pakety tokov do pamäte RAM, začne v tomto




Obr. 3.9: Časová os systému spätného záchytu.

momente vyhľadávať jednotlivé pakety na základe podozrivej IP adresy a ukladať ich spolu s ostatnými paketami z *Live Capture* jednotky do súboru.[18] Takto zachytené pakety sú uložené na disk vo formáte PCAP.

Z uvedeného popisu je zrejmé, že pamäťová náročnosť riešnia závisí na parametri, ktorý v jednotke *Ring Buffer* určuje počet prvých $N$ zachytených paketov. Takisto si môžeme všimnúť, že medzi časom detekcie **t3** a začiatkom záchytu v bode **t4**, uplynie istá doba, ktorá je však z pohľadu trvania celého procesu zanedbateľná.

# Kapitola 4

# Prehľad bezpečnostných anomálii

Detekcia bezpečnostných hrozieb si vyžaduje určité postupy, ktoré vedú k získaniu dostatočného množstva dát, ktoré sú neskôr podrobené analýze. Nato aby sme boli vôbec schopní získať tieto dáta je potrebná rada nástrojov. Prvá časť tejto kapitoly sa zaoberá predstavením niektorých z týchto nástrojov.

V častiach 4.2 a 4.3 budú uvedené príklady detekcie bezpečnostných hrozieb založených na analýze NetFlow dát. Taktiež bude popísaný postup akým spôsobom je možné v týchto dátach nájsť hrozby, identifikovať ich kľučové prvky, ktorými sa odlišujú od bežnej komunikácie a na základe týchto prvkov navrhnuť prípadné opatrenia k ich zamedzeniu.

## 4.1 Nástroje použité pri analýze

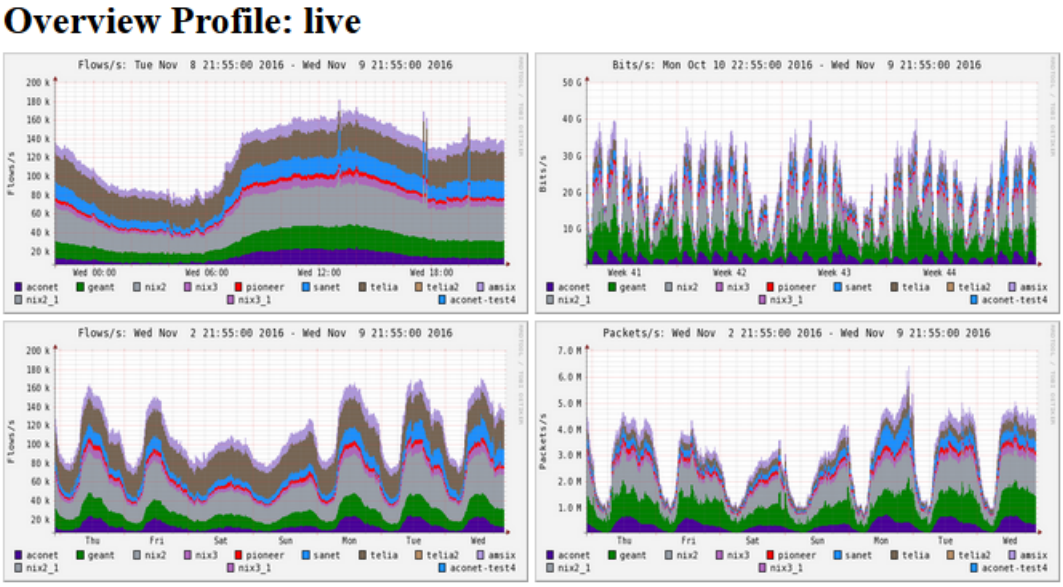
Počiatočným bodom pri detekcii bezpečnostných hrozieb je schopnosť vedieť získať relevantné informácie, ktoré tvoria základ celej analýzy. Pri dnešnom objeme komunikácie sa množstvo zachytávaných NetFlow dát pohybuje v rádoch stoviek gigabajtov, z čoho vyplýva čoraz väčší dôraz na efektívne získavanie informácií z týchto dát. Práve na tento účel existuje rada nástrojov, ktoré sú schopné filtrovať NetFlow dáta na základe rôznych parametrov. Jedným z nich je aj nástroj **Nfdump**, ktorý som práve pri tejto práci využil.

Nfdump je software, ktorý obsahuje sadu niekoľkých nástrojov, ktoré dokážu spracovávať a filtrovať Netflow dáta. Výstupom po aplikácii filtra sú štatistiky jednotlivých tokov, ktoré je možné vidieť na obrázku 4.1.

| Date first seen | Duration | Proto | Src IP Addr:Port | | Dst IP Addr:Port | Flags | Tos | Packets | Bytes | Flows |
|---|---|---|---|---|---|---|---|---|---|---|
| 2016-11-01 10:24:29.199 | 0.000 | TCP | Src IP:20702 | -> | Dst IP:23 | ....S. | 0 | 1 | 40 | 1 |
| 2016-11-01 10:24:25.166 | 0.000 | TCP | Src IP:46815 | -> | Dst IP:1080 | .A.... | 0 | 1 | 41 | 1 |
| 2016-11-01 10:24:25.541 | 0.000 | TCP | Src IP:23722 | -> | Dst IP:2323 | ....S. | 0 | 1 | 40 | 1 |
| 2016-11-01 10:24:24.047 | 0.000 | TCP | Src IP:21320 | -> | Dst IP:50814 | .A.... | 0 | 1 | 40 | 1 |
| 2016-11-01 10:24:04.419 | 21.455 | TCP | Src IP:9050 | -> | Dst IP:53536 | .AP... | 0 | 46 | 66148 | 1 |
| 2016-11-01 10:24:19.410 | 0.000 | TCP | Src IP:23 | -> | Dst IP:62354 | .A.R.. | 0 | 1 | 40 | 1 |
| 2016-11-01 10:24:29.491 | 0.000 | TCP | Src IP:36754 | -> | Dst IP:23 | ....S. | 0 | 1 | 40 | 1 |
| 2016-11-01 10:24:22.086 | 5.206 | TCP | Src IP:28095 | -> | Dst IP:23 | ....S. | 0 | 2 | 80 | 1 |
| 2016-11-01 10:24:22.518 | 0.426 | TCP | Src IP:18952 | -> | Dst IP:80 | .AP.SF | 0 | 5 | 627 | 1 |
| 2016-11-01 10:24:26.592 | 0.000 | TCP | Src IP:37877 | -> | Dst IP:23 | ....S. | 0 | 1 | 44 | 1 |
| 2016-11-01 10:23:21.609 | 61.186 | TCP | Src IP:33533 | -> | Dst IP:443 | .AP.SF | 0 | 15 | 2269 | 1 |
| 2016-11-01 10:24:25.745 | 0.000 | TCP | Src IP:45618 | -> | Dst IP:1911 | ....S. | 0 | 1 | 40 | 1 |

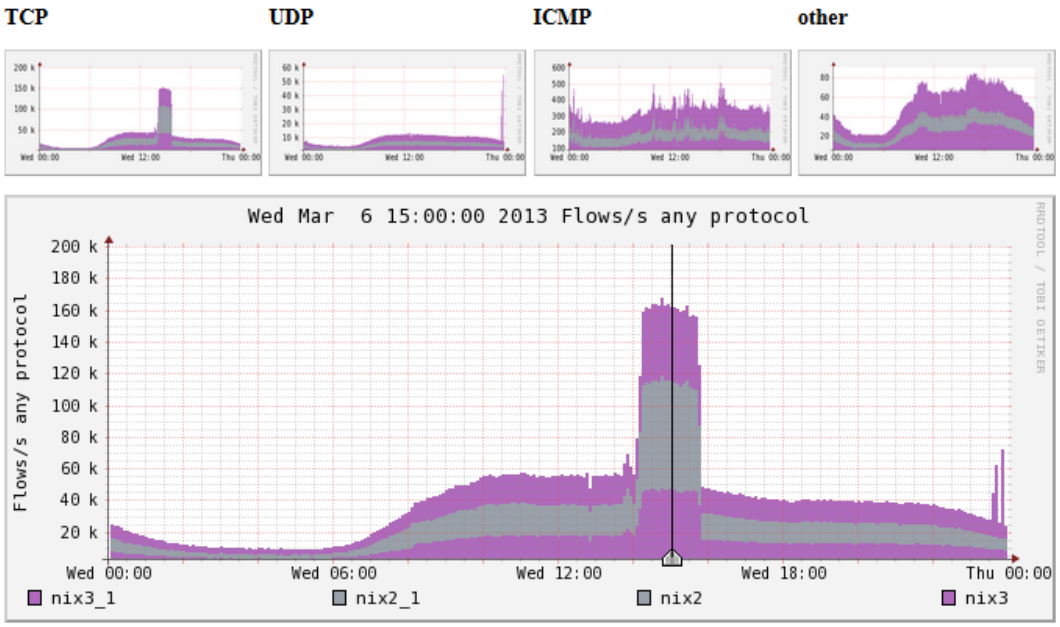
Obr. 4.1: Formát výstupu nástroja Nfdump.

Ďalším z použitých nástrojov je NfSen, ktorý slúži ako GUI samotného Nfdumpu. Tento nástroj umožňuje rýchle a intuitívne zobrazovanie NetFlow štatistík v podobe grafov, ktoré je možné vidieť na obrázku 4.2.




Obr. 4.2: Zobrazenie štatistík v nástroji NfSen.

Prepojenie s Nfdumpom umožnuje vytvárať rôzne druhy filtrov a výsledky zobrazovať priamo v prehliadači. Tieto filtre môžu pomôcť ku identifikácii niektorých anomálií ako napríklad DDoS útok, ktorý sa môže prejavovať v grafe v podobe ilustrovanej na nasledujúcom obrázku:




Obr. 4.3: DDoS útok v NetFlow dátach.

## 4.2 Útok hrubou silou

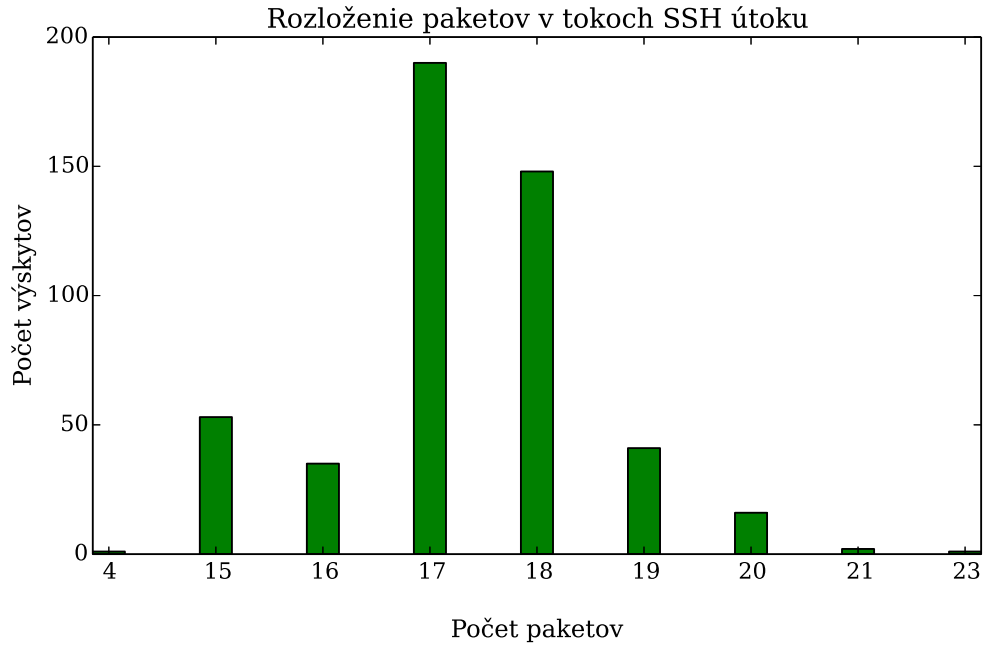
Jedným z často detegovaných útokov je útok hrubou silou (*angl. Brute force attack*). Tento typ útoku má veľmi jednoduchú podstatu, avšak pri profesionálnom prevedení dokáže napáchať nemalé škody. Princípom je automatické hádanie hesiel na určitý typ služby, o ktorej útočník vie, že je na danom stroji dostupná. Zvyčajne sa stretávame s typmi služieb ako SSH, RDP, SMB alebo Telnet.

Realizácia útoku na tieto služby spočíva v generovaní veľkého množstva spojení za účelom uhádnuť heslo a tým dostať cieľový stroj pod svoju kontrolu. V tomto prípade sa často strávame s použitím voľne dostupných slovníkov hesiel, pomocou ktorých sa dá útok zefektívniť.

Automatizované prevedenie je však jednou z kľučových vlastností, ktorá sa využíva pri jeho detekcii, nakoľko takto generovaný útok vytvára určitú signatúru na základe ktorej je možné postaviť detekčný algoritmus. Detekcia môže byť teda založená na anlyzovaní zachytených dát, ktoré obsahujú útok a následnej extrakcii odpovedajúcej signatúry, ktorej hlavnými prvkami je štatistika počtu bajtov a paketov v jednotlivých tokoch.

Nad získanými štatistikami je možné následne vykonávať rôzne druhy operácií. Jednou z nich je napríklad vytvorenie histogramu, ktorý zachytáva rozloženie kľúčových štatistík v grafickej podobe. Pomocou takto zostrojeného histogramu je možné detegovať obecnejšie útoky, bez nutnosti ručného nastavovania niekoľkých detekčných prahov.

Na nasledujúcom obrázku je možné vidieť histogram zachytávajúci rozloženie počtu paketov v tomto type útoku na protokol SSH:




Obr. 4.4: Príklad histogramovej signatúry útoku na protokol SSH.

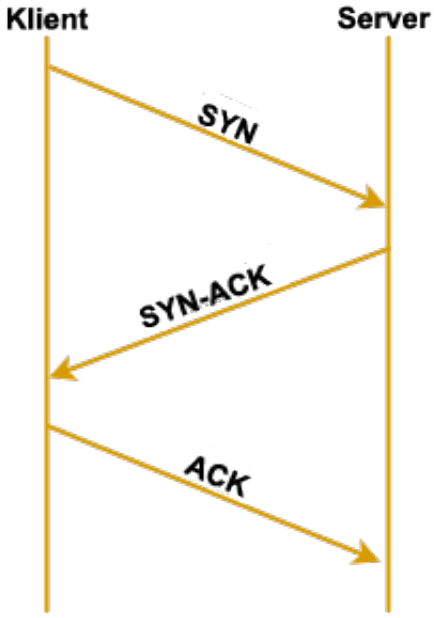
## 4.3 DDoS útok

Ďalším z populárnych útokov je DDoS (*Distributed Denial of Service*). Tento útok je založený na princípe znemožnenia komunikácie medzi užívateľom a cieľovým serverom, na ktorom beží určitý typ služby, ktorú užívateľ používa.

Na začiatku sa útočník snaží získať veľké množstvo počítačov, ktoré použije na generovanie útoku. Väčšinou sa jedná o určitý typ škodlivých programov tzv. *malware*, ktoré sú schopné bez vedomia majteľa vykonávať sady príkazov na pokyn útočníka. Takto napadnuté stroje potom tvoria prepojenú sieť tzv. *botnet*, ktorá je použitá na realizovanie samotného útoku.

Následne je vykonaný útok, kedy napadnuté počítače začnú generovať veľké množstvo spojení na cieľový server a tým môže dôjsť k jeho vyťaženiu do takej miery, že nebude schopný obsluhovať legitímne požiadavky. Takisto je možné zamerať útok napríklad na vyčerpanie pamäte cieľového stroja.

Jedným z útokov tohto typu je aj *TCP SYN Flood*. Útok využíva počiatočnú fázu nadväzovania TCP spojenia tzv. *TCP 3-way handshake*. Pri nadväzovaní legitímneho TCP spojenia, zdrojový stroj zasiela prvý paket s nastaveným SYN príznakom, načo cieľový stroj odpovedá potvrdením vo forme SYN-ACK paketov. Ak stroj, ktorý zahajoval spojenie príjme tieto pakety, potvdí ich cieľovému stroju pomocu ACK paketu. Od tohto momentu sa považuje TCP spojenie za nadviazané.[12] Postup nadväzovania je možné vidieť na nasledujúcom obrázku:




Obr. 4.5: TCP 3-way handshake.

Princípom tohto útoku je nezaslanie potvrdzujúceho ACK paketu na cieľový server, prípadne sa stretávame s variantou kedy je v prvom SYN pakete podvrhnutá cieľová IP adresa tzv. *spoofing*. To má za následok, že server pošle dvojicu SYN-ACK paketov na IP adresu, ktorá nikdy neposlala úvodný SYN a teda nemá dôvod odpovedať na tieto pakety pomocou ACK paketu. Cieľový server si však na toto čiastočne otvorené spojenie vyhradí určité množstvo zdrojov. Pokiaľ je počet takto iniciovaných spojení dostatočne

veľký, postupom času dochádza k vyčerpaniu dostupných zdrojov na serveri čo môže viesť k jeho zlyhaniu a v konečnom dôsledku k neschopnosti odpovedať na legitímne požiadavky od užívateľov.

Základnými časťami algoritmu sú detekčné prahy, ktoré slúžia na porovnávanie so štatistikami zbieranými z komunikácie. Ide o sadu hodnôt, ktoré špecifikujú hraničnú hodnotu pre každú sledovanú štatistiku. V prípade, že ukladaná štatistika prekročí tento prah, je označená za podozrivú. V závislosti na detekčnom algoritme môže dôjsť priamo ku generovaniu výstrahy, prípadne je možné priradiť tomuto prekročeniu istú váhu, ktorá je neskôr sčítaná s váhami ostatných štatistík a v prípade prekročenia medznej hodnoty dochádza ku generovaniu výstrahy.

Pri tomto type útoku dochádza k detekcii ako útočníka tak aj obete útoku vzhľadom na nadmerný objem komunikácie na zdrojovú, či cieľovú IP adresu. Ďalej sú zbierané základné štatistiky o tokoch, ako napríklad počet paketov a bajtov. Zbierané údaje slúžia práve pre porovnávanie oproti detekčným prahom. Taktiež sa do detekčných pravidiel môžu zahrnúť obmedzujúce podmienky, ako napríklad rôzne nastavenie TCP príznakov.

# Kapitola 5

# Návrh tvorby mitigačných pravidiel

Nasledujúca kapitola popisuje návrh tvorby mitigačných pravidiel pre systém spätného záchytu. Cieľom návrhu je vytvorenie charakteristiky pre DDoS útok zo zachytených dát, ktorá umožní pokryť čo možno najväčšie množstvo nežiadúcej komunikácie. Výsledok návrhu by mal zohľadňovať počet vytvorených pravidiel tak, aby ich aplikovanie bolo realizovateľné. Výsledkom celej metódy by mala byť sada položiek, z ktorých je možné zostavit mitigačné pravidlá, ktoré po aplikovaní dokážu zmierniť rozsah prebiehajúceho DDoS útoku a tým minimalizovať jeho dopad.

V prvej časti budú popísané základné zložky mitigačného pravidla a ich význam. Bude načrtnutý postup pri analýze získaných dát, ktorej hlavným cieľom bolo získanie znalostí o tom, aké hodnoty zo zachytených dát je možné použiť pri tvorbe mitigačných pravidiel. Časť 5.2 sa zaoberá samotným návrhom jednotlivých častí metódy.

## 5.1 Základné zložky mitigačných pravidiel

Predpokladom pre vytvorenie metódy umožňujúcej mitigáciu DDoS útoku pomocou pravidiel bol fakt, že sa prebiehajúci útok odlišuje v určitej miere od normálnej komunikácie. Na základe toho bolo vytvorených niekoľko detekčných algoritmov, implementovaných v rámci modulov v systéme NEMEA, ktoré však väčšinou pracujú nad NetFlow dátami. Keďže bude táto metóda pracovať nad dátami uloženými vo forme PCAP súborov, je možné očakávať, že v rámci týchto dát bude opäť možné identifikovať určité špecifické vlastnosti na základe ktorých bude možné odlíšiť prebiehajúci útok od legitímnej komunikácie.

V tomto prípade sa však nedá hovoriť o vytvorení učitého druhu signatúry útoku, nakoľko nieje možné vo všetkých prípadoch jednoznačne určiť, či vybraná zložka patrí do signatúry alebo nie. Z toho dôvodu bude prístup pri návrhu metódy zvolený tak, aby nedochádzalo k obmedzovaniu zložiek mitigačného pravidla len na určitý druh alebo ich počet.

Pre tvorbu pravidiel sa v tomto prípade budú používať informácie obsiahnuté v IP a TCP vrstvách paketu. V prípade útoku na protokol HTTP je možné tiež využiť položky z hlavičiek HTTP požiadavku na server, ktoré sú v prípade niektorých nástrojov generujúcich DDoS útok často upravované. Prehľad analyzovaných položiek je možné vidieť v nasledujúcej tabuľke:

| Type | Items |
|---|---|
| IP | DSCP(ToS), Total length, Identification, Flags, Fragment offset, TTL, Header checksum |
| TCP | Sequence number, Acknowledgment number, Data offset, Flags, Window size, Checksum, Urgent pointer |
| HTTP | Accept-encoding, Accept-charset, Connection, Origin, User-agent, Cache-control, Via, Accept-language, ... |

Tabuľka 5.1: Prehľad analyzovaných položiek.

Súčasťou návrhu metódy je stanovenie hypotézy, ktorá predpokladá využitie ľubovolnej položky z vyššie spomenutých vrstiev v prípade, že je pomocou nej aspoň čiastočne možné označiť určitú časť útoku. Táto hypotéza sa však počas testovacej fázy ukázala len ako čiastočne pravdivá. Dôvodom bolo zistenie, že v špecifických prípadoch môžu niektoré položky vykazovať hodnoty, ktoré sa výrazne líšia od bežných hodnôt vyskytujúcich sa v legitímnej komunikácii, avšak nemusí sa nutne jednať o útok.

## 5.2 Návrh vytvárania pravidiel

Proces tvorby mitigačných pravidiel bude rozdelený na dve základné časti. V prvej časti bude prebiehať detekcia DDoS útokov prostredníctvom NEMEA modulu. Pri detekovaní útoku sa vygeneruje výstraha, ktorá bude spracovaná a na jej základe sa pomocou ďalšieho modulu aktivuje zachytávanie celej komunikácie pomocou systému *Time machine*, ktorého architektúra a princíp fungovania boli rozobraté v kapitole 3.4.4.

Druhou časťou bude vlastné fungovanie generátoru migitačných pravidiel. Ten dostane upozornenie o tom, že bol zachytený DDoS útok a následne si vyhľadá v príslušnom dátovom úložisku dáta prislúchajúce tomuto útoku. Analýza prebehne nad dátami uloženými vo forme PCAP súborov. Na obrázku 5.1 je možné vidieť celý proces od vzniku útoku až po vytvorenie mitigačného pravidla.

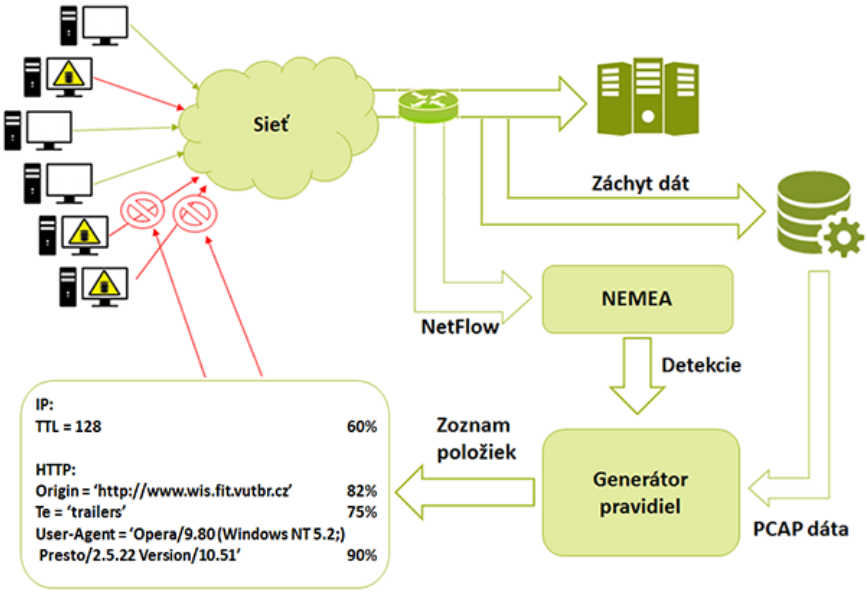
Druhú fázu, ktorej implementáciou sa zaoberá kapitola 6 je možné rozdeliť na menšie vzájomne prepojené časti, ktoré sa vykonávajú pre oba vstupné PCAP súbory. Týmito časťami sú:

- Extrakcia dát
- Výpočet štatistík
- Porovnanie profilov
- Overenie položiek

Výsledkom porovnávania profilov komunikácie bude zoznam položiek, ktoré sú dostatočne význačné, aby bolo pomocou nich možné odlíšiť prebiehajúci útok od bežnej komunikácie. Problémom je, že v niektorých prípadoch sa môže v tomto zozname vyskytnúť aj položka, ktorá po zakomponovaní do mitigačného pravidla výrazne ovplyvní rozsah označených dát, prípadne označí dáta, ktorých časť nemusí nutne patriť do dát útočiacej komunikácie. Príčinou zaradenia tejto položky do zoznamu položiek pre mitigačné pravidlo môžu byť napríklad špecifické dáta v súbore obsahujúcom útok.

Na základe toho po vykonaní porovnania profilov prebehne overenie položiek, ktoré budú vybrané zo vstupných dát. Princíp overovania položiek bude spočívať v tom, že pre

každú položku bude vypočítaná hodnota pokrytia v rámci útočiacej datovej sady. Táto hodnota bude uvedená v percentách a v prípade, že sa administrátor rozhodne vytvoriť mitigačné pravidlo, môže sa riadiť práve pomocou tejto hodnoty.




Obr. 5.1: Celkový pohľad na proces tvorby mitigačných pravidiel.

# Kapitola 6

# Implementácia navrhnutej metódy

V tejto kapitole bude popísaný spôsob realizovania navrhnutej metódy a zároveň budú postupne objasnené detaily jednotlivých jej častí. Podkapitola 6.1 sa zaoberá princípom extrahovania dát zo vstupných súborov. V časti 6.2 bude popísaný princíp výpočtu štatistík zo sledovaných položiek a v časti 6.3 bude detailne objasnený pospup pri porovnávaní týchto štatistík, ktoré reprezentujú zostavený profil komunikácie. Záver kapitoly sa venuje poslednej časti navrhnutej metódy, ktorou je proces overovania vybraných položiek pre mitigačné pravidlo.

## 6.1 Extrakcia dát

V prvej časti dochádza k extrakcii dát zo vstupných súborov. Generátor teda pracuje s dvoma súbormi. Prvý súbor obsahuje dáta, ktoré sa považujú za dáta legitímnej komunikácie. Keďže súčasťou výstrahy vygenerovanej systémom NEMEA je aj IP adresa napadnutej stanice, je možné sa priamo zamerať len na dáta, ktoré popisujú komunikáciu tejto stanice. Obsahom druhého súboru je okrem iného aj komunikácia obsahujúca prebiehajúci útok na cieľovú stanicu. Z oboch súborov sa extrahujú vybrané položky, ktorých zoznam je možné vidieť v tabuľke 5.1.

## 6.2 Výpočet štatistík

Po spracovaní celej komunikácie je potrebné vytvoriť štatistiky sledovaných položiek. Tieto štatistiky sa odlišujú v závislosti od typu položky. V prípade že sledovaná položka nadobúda rôzne hodnoty, z ktorých má zmysel počítať štatistiky, je pre túto položku vypočitaná jej stredná hodnota, odchýlka od strednej hodnoty a najpočetnejšie zastúpená hodnota. Medzi tento druh položiek patrí napríklad **Sequence number**, **Window size**, **Header checksum** a iné.

V iných prípadoch, kedy by tieto štatistické hodnoty vzhľadom na sémantiku položky nedávali žiadny zmysel, dochádza k výpočítaniu percentuálneho zastúpenia jednotlivých hodnôt v rámci celej množiny. Výsledkom je teda zastúpenie každej hodnoty v uložených dátach vyjadrené v percentách. Tento typ je uplatnený napríklad v prípade TCP príznakov.

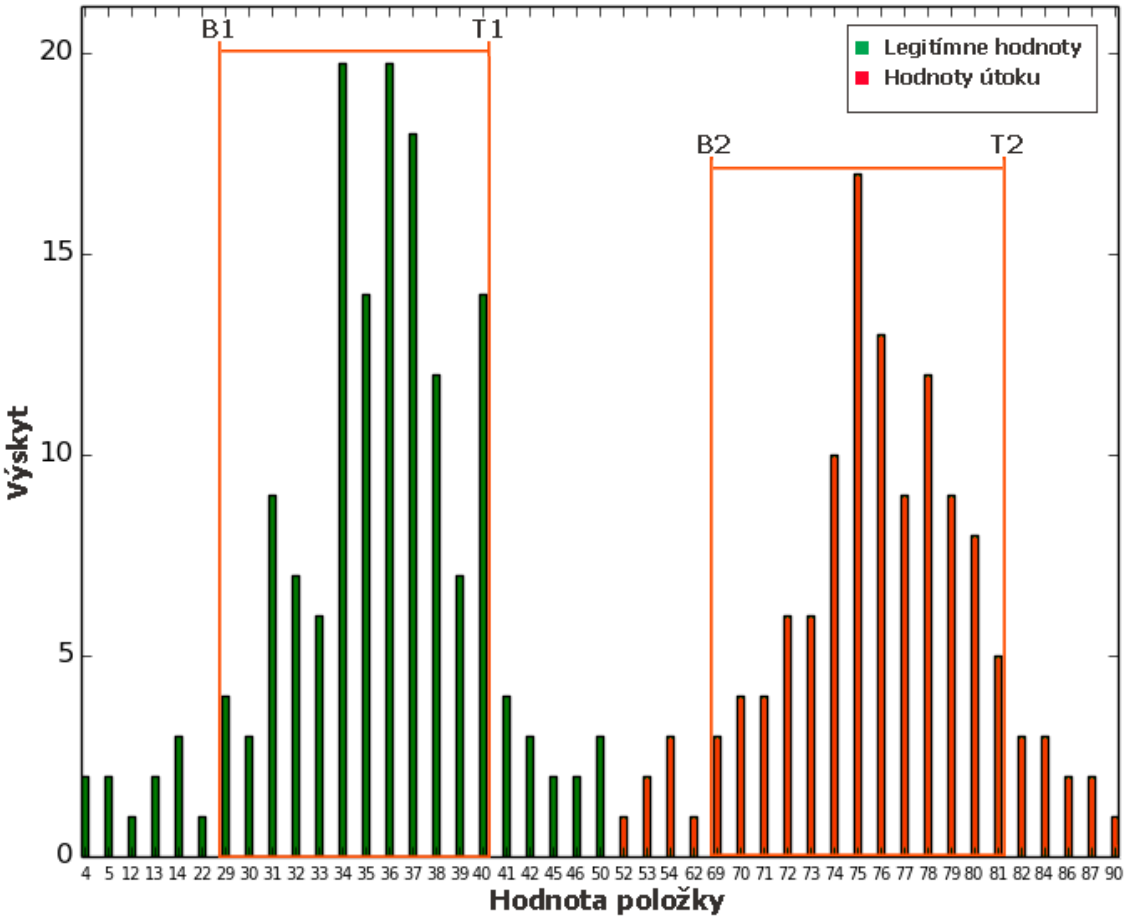
Po vypočítaní percentuálneho zastúpenia sa uloží hodnota s najvyšším percentuálnym zastúpením. Tým pádom sa predchádza zbytočnému ukladaniu ostatných percentuálnych hodnôt, ktorých počet by mohol byť veľký v prípade, ak by sa celkové rozloženie hodnôt približovalo uniformnému rozloženiu. Navyše by tieto hodnoty mali veľmi zanedbatelný dopad

na označenie útočiacich paketov. Túto situáciu je možné sledovať napríklad pri položkách z HTTP hlavičky.

Po získaní štatistík pre všetky sledované položky sa vytvorí tzv. profil komunikácie, ktorý tvorí súhrnný pohľad na celú komunikáciu prostredníctvom získaných štatistík o sledovaných položkách. Po zostavení oboch profilov, dochádza k záverečnej fáze porovnávania, ktorej výsledkom sú už samotné položky mitigačného pravidla.

## 6.3 Porovnanie profilov

Poslednou fázou je porovnávanie zostavených profilov komunikácie. Porovnávanie prebieha pomocou porovnávacích pravidiel, na základe ktorých sa určuje, či daná položka patrí do mitigačného pravidla. V prípade, že sa jedná o položku s bežnými štatistickými hodnotami je porovnávanie vykonané metódou, ktorú je možné chápať ako porovnávanie dvoch rámcov, vytvorených zo štatistických hodnôt tejto položky. Nasledujúci obrázok ilustruje základný druh porovnania:




Obr. 6.1: Ukážka jednoduchého porovnania.

Pre zjednodušenie sú hodnoty v grafe upravené tak, aby bolo možné demonštrovať princíp jednotlivých porovnaní. Na obrázku je možné vidieť dve odlišné množiny hnodnôt pre danú položku. Na začiatku porovnania sa vytvorí rámec pre každú množinu hodnôt.

Následne sa zisťuje vzájomná poloha týchto dvoch vytvorených rámcov. Dolná hodnota rámca je vypočítaná odčítaním smerodatnej odchýlky od strednej hodnoty a analogicky horná hodnota rámca je vypočítaná pričítaním smerodatnej odchýlky k strednej hodnote.

Rámec vyvtorený z hodnôt legitímnej komunikácie je na obrázku reprezentovaný hraničnými hodnotami **B1** a **T1**. Rámec reprezentovaný hraničnými hodnotami **B2** a **T2** patrí hodnotám položky z útočiacej komunikácie.

Pokiaľ sa porovnaním zistí, že sa oba rámce neprekrývajú, sú hodnoty reprezentované útočiacim rámcom považované za dostatočne odlišné od hodnôt sledovanej položky legitímnej komunikácie. Tým pádom je táto položka pridaná do zoznamu položiek pre mitigačné pravidlo.

Hodnota tejto položky môže nadobúdať presnú hodnotu alebo môže ísť o interval vymedzujúci určité hodnoty vzhľadom na vzájomnú pozíciu najpočetnejšej hodnoty a rámca [**B2**;**T2**]. Pokiaľ sa táto hodnota nachádza vo vnútri rámca, je priamo vybraná a bude použitá v mitigačnom pravidle. V prípade, že sa hodnota nenachádza vo vnútri rámca, je pre položku mitigačného pravidla zvolený interval, vytvorený z hodnôt utočiaceho rámca.

Pri pohľade na obrázok 6.1 je možné vidieť, že tento druh rozloženia rámcov sa vyskytuje v prípade, že je niektorá z položiek útoku výrazne odlišná od položky v bežnej komunikácii. V konečnom dôsledku je teda výhodou, pokiaľ sa útočník snaží útočiť pomocou nástroja, ktorý generuje unikátne hodnoty niektorých položiek, nakoľko sú tieto hodnoty ľahko detegovateľné. V praxi sa samozrejme stretávame s rôznym rozložením hodnôt položiek, čo sa v uvedenej ilustračnej reprezentácii prejaví ako čiastočné prekrytie oboch rámcov. V takom prípade dochádza k vypočítaniu prekrytia medzi rámcami, ktoré je opäť možné chápať ako spôsob, ktorým je možné zistiť mieru odlišnosti bežnej a útočiacej vzorky dát.

Prienik útočiaceho rámca s rámcom bežnej komunikácie je daný percentuálnou hodnotou. Pokiaľ sa rámce prerkývajú na menej ako 30% plochy, je položka pridaná do zoznamu položiek pre mitigačné pravidlo. Hodnotou položky je v tomto prípade priamo interval reprezentovaný útočiacim rámcom. Percentuálna hodnota prekryvu, ktorá rozhoduje o zaradení položky do mitigačného pravidla, bola určená na základe vykonaných experimentov a jej hodnotu je možné meniť.

Pokiaľ je prekryv rámcov väčší ako 30%, prípadne sa rámce prekrývajú úplne, položka nieje zaradená do mitigačného pravidla, nakoľko sa hodnoty útočiacej položky výrazne neodlišujú od hodnôt bežnej komunikácie a aplikovaním pravidla obsahujúceho takúto položku by s veľkou pravdepodobnosťou došlo aj k označeniu legitímnej komunikácie.

Počas vykonávania experimentov, ktorým sa venuje kapitola 7 boli zistené ďalšie možné kombinácie rozložení hodnôt v rámci porovnávaných položiek. Na základe týchto poznatkov, bola metóda rozšírená o niekoľko ďalších optimatizácií, ktoré zrýchľujú proces porovnávania, prípadne sú aplikované v špeciálnych prípadoch.

Príkladom je vytvorenie doplňujúceho pravidla pre porovnávanie, ktoré je aplikované v prípade, že rozloženie hodnôt sledovanej položky z legitímneho profilu má charakter uniformného rozloženia. Pokiaľ má rozloženie hodnôt položky z útočiaceho profilu neuniformný charakter, prípadne sú všetky hodnoty položky rovnaké je možné túto položku zaradiť do zoznamu pre mitigačné pravidlo. Jej hodnota môže byť opäť repreznovaná intervalom hodnôt, prípadne môže ísť o jednu presnú hodnotu. V oboch prípadoch sú hodnoty odvodené z hodnôt vybranej položky útočiaceho profilu.

Jednou z ďalších úprav, ktoré boli vykonané na základe dodatočnej analýzy vytvorených profilov, bolo pridanie pravidla, ktoré má za úlohu odhaliť upravenú hodnotu sledovanej položky. Na základe vyššie uvedeného princípu porovnávania, ktoré je možné chápať ako porovnávanie dvoch rámcov dát zobrazených v histograme je zrejmé, že s rastúcou podob-

noťou hodnôt sledovanej položky, by malo dochádzať aj k väčšiemu prekryvu rámcov. Z toho vyplýva, že popisovaná metóda vyhodnotí položku ako nevhodnú pre zaradenie do mitigačného pravidla.

Pri bližšej analýze útočiacich dát bolo zistené, že v prípade ak útočník pozmení hodnotu niektorej z položiek výrazným spôsobom a zároveň sa bude snažiť túto zmenu maskovať generovaním bežnej hodnoty tejto položky je možné, že sa bude rámec vypočítaný z týchto štatistík dostatočne prekrývať s hodnotami bežného rámca. Avšak početný výskyt hodnoty pozmenenej položky, ktorá má potenciál identifikovať aspoň určitú časť útočiacich paketov, nebude braný v tomto prípade v úvahu. Túto anomáliu si je možné jednoducho predstaviť ako výraznú špičku, ktorá sa vyskytuje mimo hlavnej časti dát zobrazených v histograme.

Z toho dôvodu sa pri tvorbe profilu, ukladá štatistika o najpočetnejšej hodnote vrámci skúmanej položky. Pokiaľ sa počas porovnávania rámcov zistí, že ich prekryv prekračuje určitú hranicu, dochádza k dodatočnej kontrole, v ktorej sa zisťuje pozícia najpočtnejšej hodnoty položky z útočiaceho profilu. V prípade, že sa nachádza mimo rámec legitímnej komunikácie, je táto položka zaradená do zoznamu položiek pre mitigačné pravidlo s hodnotou najpočetnejšej položky.

Ak je pre analyzovanú položku v zostavenom profile uložená štatistika vo forme percentuálneho zastúpenia hodnôt, vykonáva sa porovnanie na základe tejto percentuálnej hodnoty. Z útočiaceho profilu je vybraná hodnota položky, ktorá má najväčšie percentuálne zastúpenie v rámci množiny hodnôt. V profile legitímnej komunikácie sa pre danú položku taktiež nájde prislúchajúca percentuálna hodnota. Tieto dve hodnoty sa následne porovnávajú.

Porovnávanie prebieha opäť vo viacerých krokoch, kedy sa na začiatku zisťuje, či vôbec hodnota položky vybraná z útočiaceho profilu existuje vrámci profilu bežnej komunikácie. Pokiaľ sa táto hodnota položky v profile nenachádza, je útočiaca položka spolu so svojou hodnotou zaradená do zoznamu položiek mitigačného pravidla. V opačnom prípade je jej percentuálna hodnota porovnaná s hodnotou položky uloženej v profile bežnej komunikácie. V prípade, že je rozdiel oboch percentuálnych zastúpení väčší ako definovaná hranica, je útočiaca položka spolu s jej hodnotou zaradená do zoznamu pre mitigačné pravidlo.

Tento druh porovnania sa používa napríklad pri TCP príznakoch alebo položkách extrahovaných z HTTP hlavičiek. Hraničná hodnota použitá v porovnaní bola opäť zvolená na základe výsledkov experimentov a je možné meniť jej hodnotu.

## 6.4 Overenie položiek mitigačného pravidla

Porovnávanie položiek je realizované bez použitia statickej šablóny, ktorá by obsahovala zoznam položiek, ktoré sa majú porovnávať za každých okolonností. Z toho vyplýva, že pokiaľ vzorka dát obsahuje okrem útoku aj veľké množstvo inej komunikácie je možné, že niektoré položky, ktoré majú potenciál identifikovať útok nebudú v tomto prípade označené ako vhodné položky do mitigačného pravidla, prípadne pokiaľ budú označené ako vhodné, ich hodnoty môžu byť ovplyvnené touto komunikáciou. V takom prípade je zaradenie tejto položky do pravidla určitým rizikom, nakoľko na základe jej hodnoty môže byť označená legitímna časť komunikácie.

Z toho dôvodu bola metóda rozšírená o overovanie položiek, ktoré má za účel pre každú položku mitigačného pravidla určiť jej dopad na datovú sadu obsahujúcu útok. Pomocou každej položky je vykonané filtorvanie datovej sady a na základe počtu odfiltorvaných dát je vypočítaný dopad tejto položky na útočiace dáta v percentách. Výsledkom tohto testu,

je usporiadanie vybraných položiek na základe ich percentuálnej hodnoty, čo je v konečnom dôsledku aj výstupom celej metódy.

# Kapitola 7

# Experimenty a testovanie

Obsahom tejto kapitoly je popis vykonaných experimentov nad reálnymi dátami za účelom zistenia funkčnosti navrhnutej metódy. Jedným z hlavných cieľov experimentov bola snaha o zistenie do akej miery sú výsledky metódy ovplyvnené obsahom datovej sady reprezentujúcej útok. Práve z toho dôvodu boli jednotlivé experimenty opakovane vykonávané nad rôzne upravenou datovou sadou obsahujúcou útok.

V časti 7.1 je objasnený postup získavania testovacích dát, ktoré tvorili základ pre jednotlivé experimenty. Získané dáta slúžili ako vstup do generátoru mitigačných pravidiel. Princíp jednotlivých experimentov a ich výsledky popisuje časť 7.2. Časť 7.3 sa zaoberá alternatívnym prístupom pri analýze dát pomocou nástroja Weka, ktorý umožňuje vykonávať klasifikáciu zozbieraných dát do definovaných tried a záverečná časť sa venuje celkovému zhrnutiu dosiahnutých výsledkov.

## 7.1 Postup získavania testovacích dát

Na začiatok bolo potrebné získať relevantné dáta z ktorých by bolo možné extrahovať potrebné informácie na vytvorenie základnej charakteristiky pre mitigačné pravidlá. Na tento účel bol použitý detektor *HostStats* z balíku NEMEA. Konfigurácia detektoru bola upravená tak, aby sa zameriaval na detekciu SYN flood DDoS útokov. Na základe detekcií, ktoré boli výstupom tohto modulu, boli pomocou dodatočnej analýzy nájdené dáta obsahujúce reálne útoky.

Ďalším zdrojom dát boli experimenty, ktoré pozostávali z generovania útokov na lokálny stroj, na ktorom zároveň prebiehal záchyt dát pomocou nástroja Wireshark. Cieľom týchto experimentov bolo hlavne získanie rôznorodej sady dát určenej k analýze. Každý z experimentov používal iný nástroj pre generovanie útokov. Na internete je dostupných mnoho nástrojov, ktoré sú schopné generovať DDoS útoky dokonca aj bez hlbšej znalosti problematiky.

V rámci každého experimentu bolo pomocou zvoleného nástroja vykonaných niekoľko útokov, pričom pri každom z nich bola aspoň čiastočne pozmenená konfigurácia, s ktorou bol nástroj spustený. Výber bol cielene zameraný na pokrytie väčšiny nástrojov od tých najjednoduchších, ktoré sú ľahko konfigurovateľné a teda vžadujú len veľmi málo znalostí danej problematiky až po tie, ktoré sa radia do skupiny pokročilejších, nakoľko je pomocou nich možné generovať špecifické druhy útokov.

Nástroje použité na generovanie DDoS útokov v rámci experimentov:

- LOIC - Aplikácia, ktorá je pomerne rozšírená hlavne pre jej grafické rozhranie a jednoduchosť ovládania, bez nutnosti ďalšieho konfigurovania. Umožňuje zvoliť IP adresu a port cieľovej stanice, vybrať typ tansportného protokolu, prípadne nastaviť špecifickú správu, ktorá bude obsahom paketu.
- Thors Hammer - Jednoduchý skript, ktorý umožňuje DDoS útok na ľubovolnú adresu, bez dotatočnej konfigurácie.
- Hulk - Python skript, ktorý umožňuje priamo meniť typ a vlastnosti útoku ako aj hodnoty zasielaných parametrov. Tým pádom je pomocou neho možné vytvoriť pakety, ktorých položky môžu obsahovať ľubovolné hodnoty.

Pomocou uvedených nástrojov boli vykonané DDoS útoky na protokol TCP a webový server, na ktorom bežala služba HTTP. Datové sady použité pre generátor mitigačných pravidiel boli vytvorené tak aby obsahovali určitý pomer útočiacich a legitímnych dát.

Prvá datová sada obsahuje záchyt len samotného útoku. Druhú datovú sadu tvoria pakety zachyteného útoku spolu s paketami legitímnej komunikácie, ktorá bola simulovaná počas priebehu útoku. Dáta obsahujúce útok sú pomere 2:1 oproti dátam legitímnej komunikácie. Obsahom tretej datovej sady sú opäť oba druhy komunikácie, tentokrát v pomere 1:1. V poslednej datovej sade objem legitímnej komunikácie výrazne prevyšuje objem dát útočiacej komunikácie.

Táto sada bola špeciálne vytvorená pre zistenie efektivity navrhnutej metódy v prípade, že sa v zachytených dátach nachádza väčšie možstvo legitímnej komunikácie než útočiacej. Tento prípad by však nemal nastávať často, keďže pri výskyte DDoS útoku na špecifickú IP adresu, sa celkový objem komunikácie zvýši práve o komunikáciu, ktorá tvorí útok. Tým pádom by mal byť objem zachytených dát tvorený z väčšej časti práve dátami útoku.

Opačná situácia by mohla nastať v prípade, že počas útoku by bola zároveň generovaná aj legitímna komunikácia na cieľovú IP adresu, či už priamo za účelom maskovania prebiehajúceho útoku, alebo by sa jednalo o bežnú komunikáciu cieľovej stanice. V takom prípade je zaujímavé sledovať akým spôsobom sa navrhnutá metóda zachová.

Keďže navrhnutá metóda pracuje okrem datovej sady obsahujúcej útok aj s dátami legitímnej komunikácie, bolo potrebné získať aj tento druh dát. Táto datová sada bola získaná záchytom dát zo stroja na ktorom prebiehala bežná komunikácia počas určitého časového obdobia. Pre všetky vykonané experimenty bola táto datová sada použitá ako vzorka bežnej komunikácie. V závislosti na type útoku sa však vypočítané štatistiky sledovaných položiek menia nakoľko sa mení aj objem dát, z ktorých sú počítané.

## 7.2 Výsledky experimentov

Na začiatok je potrebné uviesť profil vytvorený z dát legitímnej komunikácie. V rámci nástrojov LOIC a Thors Hammer, ktorých útok bol zameraný na protokol TCP sa ako referenčný profil legitímnej komunikácie používal profil zostavený z datovej sady, obsahujúcej len legitímnu komunikáciu.

Celkový počet paketov, z ktorých bola zostavená IP časť profilu bol **670704**, pričom z toho bolo **435016** TCP paketov, čo tvorilo približne 65%. Nasledujúca tabuľka zobrazuje jednotlivé položky profilu spolu s prvým typom štatistík, ktorých výpočet bol popísaný v kapitole 6.2:

| Type | Item | Mean | STD | Mode |
|---|---|---|---|---|
| **IP** | Header checksum | 25810 | 21038 | 0 |
| | Type of service | 1.05 | 13.08 | 0 |
| | Identification | 28889 | 18706 | 0 |
| | Fragment offset | 0 | 0 | 0 |
| | TTL | 79.2 | 34.8 | 128 |
| | Total length | 1064 | 636 | 1500 |
| **TCP** | Data offset | 5.24 | 1.21 | 5 |
| | Window size | 1439 | 5498 | 245 |
| | Checksum | 32716 | 19637 | 27031 |
| | ACK number | 1795960944 | 1274870377 | 432553121 |
| | Sequence number | 2035932494 | 1220692825 | 6575645 |

Tabuľka 7.1: Štatistické hodnoty položiek v profile legitímnej komunikácie.

Nasledujúca tabuľka obsahuje položky profilu, ktorých hodnoty sú tvorené percentuálnym zastúpením:

| Type | Item | Value |
|---|---|---|
| **IP** | More fragment | False = 100% |
| | Do not fragment | False = 47.22%, True = 52.78% |
| **TCP** | Urgent flag | False = 100% |
| | Flags | SYN = 0.7% |
| | | RST = 0.09% |
| | | ACK = 85% |
| | | ACK + FIN = 0.7% |
| | | ACK + SYN = 0.4% |
| | | ACK + RST = 0.2% |
| | | ACK + PSH = 12.9% |
| | | ACK + PSH + FIN = 0.01% |

Tabuľka 7.2: Percentuálne zastúpenie položiek profilu legitímnej komunikácie.

V nasledujúcich častiach budú uvedené výsledky aplikovania navrhnutej metódy postupne pre všetky použité nástroje. V rámci každého nástroja bude uvedený zoznam položiek pre mitigačné pravidlo spolu s ich percentuálnym ohodnotením, ktorého význam bol popísaný v kapitole 6.4.

Pre každý nástroj budú uvedené 4 experimenty, ktoré odpovedajú pripraveným datovým sadám. Jednotlivé datové sady označené ako **<názov nástroja>-DS<1-4>**, pričom poradie datových sád je nasledujúce:

1. len dáta útoku
2. dáta útoku a legitímnej komunikácie v pomere 2:1
3. dáta útoku a legitímnej komunikácie v pomere 1:1
4. dáta útoku a legitímnej komunikácie v pomere viac ako 1:3

### 7.2.1 LOIC

Ako prvá bola testovaná datová sada **Loic-DS1**, výstupom metódy bol zoznam položiek uvedených v tabuľke 7.3. Výsledky datovej sady **Loic-DS2** obsahuje tabuľka 7.4. Výsledky tretej datovej sady ilustruje tabuľka 7.5 a výsledky poslednej sady dát sú obsiahnuté v tabuľke 7.6.

| **Type** | **Item** | **Value** | **Coverage** |
|---|---|---|---|
| **IP** | Header checksum | <0, 11145> | 96% |
| | Type of service | 0 | 100% |
| | Do not fragment | True | 100% |
| | Total length | 52 | 68% |
| **TCP** | Data offset | 8 | 68% |
| | Window size | 8192 | 100% |
| | ACK number | 0 | 100% |
| | Flags | SYN | 100% |
| | Checksum | 57876 | 34% |

Tabuľka 7.3: Zoznam položiek pre datovú sadu **Loic-DS1**.

| **Type** | **Item** | **Value** | **Coverage** |
|---|---|---|---|
| **IP** | Header checksum | 0 | 39% |
| | Type of service | 0 | 98% |
| | TTL | 128 | 39% |
| | Total length | <0, 1026> | 74% |
| **TCP** | Data offset | <5.6, 8.2> | 72% |
| | Window size | 8192 | 70% |
| | ACK number | 0 | 70% |
| | Flags | SYN | 70% |
| | Checksum | 57876 | 24% |
| | Sequence number | 3855239726 | 5% |

Tabuľka 7.4: Zoznam položiek pre datovú sadu **Loic-DS2**.

| **Type** | **Item** | **Value** | **Coverage** |
|---|---|---|---|
| **IP** | Header checksum | 0 | 33% |
| | Type of service | 0 | 93% |
| | Total length | 52 | 38% |
| **TCP** | Data offset | <5.2, 7.9> | 18% |
| | Window size | 8192 | 57% |
| | ACK number | 0 | 57% |
| | Flags | SYN | 57% |
| | Checksum | 57876 | 19% |
| | Sequence number | 3855239726 | 5% |

Tabuľka 7.5: Zoznam položiek pre datovú sadu **Loic-DS3**.

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | Header checksum | 0 | 14% |
| | Type of service | 0 | 92% |
| **TCP** | Window size | 8192 | 6.8% |
| | ACK number | <2385707800, 4465659609> | 92% |
| | Checksum | 1028 | 10% |
| | Sequence number | 3855270895 | 1% |

Tabuľka 7.6: Zoznam položiek pre datovú sadu **Loic-DS4**.

V prípade poslednej datovej sady kde objem legitímnej komunikácie výrazne prevyšuje objem útočiacich dát je možné vidieť úbytok položiek pre mitigačné pravidlo ako aj zmenu v hodnotách položky **ACK number**, ktorá v tomto prípade pokrýva až 92% percent celkovej komunikácie. V tomto prípade spôsobuje táto hodnota vysokú mieru falošne pozitívnych označení. Naopak hodnota položky **Windnow size** pokrýva len 6.8%, no napriek tomu po zaradení do pravidla označí práve táto položka väčšinu útočiacich dát. Malá hodnota pokrytia je v tomto prípade logická, nakoľko sa jedná o nevyváženú datovú sadu.

Kompletný rozbor pokrytia pre každú datovú sadu, ktorý vyznikol zostavením mitigačného pravidla z niektorých navrhovaných položiek ilustruje nasledujúca tabuľka, kde stĺpec s názvom TP vyjadruje hodnotu *True positive* a stĺpec FP vyjadruje mieru *False positive*:

| Data set | Rule | TP | FP |
|---|---|---|---|
| **Loic-DS1** | Type of service = 0, Do not fragment = True<br>Window size = 8192, ACK number = 0<br>TCP Flags = SYN | 100% | 0% |
| **Loic-DS2** | Type of service = 0, Total length = <0, 1026><br>Window size = 8192, ACK number = 0<br>TCP Flags = SYN | 100% | 0% |
| **Loic-DS3** | Type of service = 0, Window size = 8192<br>ACK number = 0 | 100% | 0% |
| **Loic-DS4** | ACK number = <2385707800, 4465659609> | 0% | 100% |
| **Loic-DS4** | Type of service = 0, Window size = 8192 | 99.5% | 0.5% |

Tabuľka 7.7: Výsledky aplikovania pravidiel na datové sady.

Výber kľúčových položiek pre pravidlo bol realizovaný intuitívne na základe vypočítanej hodnoty pokrytia pre každú datovú sadu. Zistené hodnoty *True positive* a *False positive* sú pre tieto vybrané položky ideálne v prvých troch datových sadách. V prípade zvolenia inej kombinácie položiek by tieto hodnoty boli výrazne odlišné, čo je možné vidieť už pri hodnotách pokrytia položiek pre jednotlivé datové sady.

Pri poslednej datovej sade je možné vidieť akým spôsobom metóda reaguje na stav kedy datová sada obshaje výrazne väčšie množstvo legitímnej komunikácie. V tomto prípade je kľučový interval hodnôt položky ACK number, ktorý označí práve pakety legitímnej komunikácie. Oproti tomu posledné pravidlo obsahuje položku Window size, ktorá mala v tabuľke 7.6 celkové pokrytie len 6.8%, avšak práve táto položka je kľúčová z pohľadu označenia veľkej časti útoku. Z tohto je zrejmé, že účinnosť metódy závisí na kvalite datovej sady reprezentujúcej útok.

### 7.2.2 Thors Hammer

Výsledky metódy pre jednotlivé datové sady sú obsiahnuté v nasledujúcich tabuľkách:

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | Header checksum | 0 | 55% |
| | Type of service | 0 | 100% |
| | Do not fragment | True | 100% |
| | TTL | 128 | 55% |
| | Total length | 52 | 73% |
| **TCP** | Data offset | 8 | 73% |
| | Window size | <5612, 11049> | 99% |
| | ACK number | 0 | 99% |
| | Flags | SYN | 99% |
| | Checksum | 57876 | 36% |

Tabuľka 7.8: Zoznam položiek pre datovú sadu **ThorsHammer-DS1**.

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | Header checksum | 0 | 46% |
| | Type of service | 0 | 99% |
| | TTL | 128 | 46% |
| | Total length | <0, 925> | 79% |
| **TCP** | Data offset | <5.4, 9.5> | 71% |
| | Window size | 8192 | 70% |
| | ACK number | <0, 1084045001> | 92% |
| | Flags | SYN | 70% |
| | Checksum | 57876 | 26% |

Tabuľka 7.9: Zoznam položiek pre datovú sadu **ThorsHammer-DS2**.

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | Header checksum | 0 | 44% |
| | Type of service | 0 | 96% |
| | TTL | 128 | 44% |
| | Total length | <0, 972> | 74% |
| **TCP** | Data offset | <5.2, 9.8> | 64% |
| | Window size | 8192 | 64% |
| | ACK number | <0, 1292624527> | 88% |
| | Flags | SYN | 64% |
| | Checksum | 57876 | 24% |

Tabuľka 7.10: Zoznam položiek pre datovú sadu **ThorsHammer-DS3**.

V tabuľke 7.11 je opäť možné vidieť výrazný vplyv objemu legitímnej komunikácie, ktorý má za následok zníženie počtu položiek v mitigačnom pravidle a logicky aj zníženie ich celkového pokrytia.

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | Header checksum | 0 | 27% |
| | Identification | 0 | 0.5% |
| **TCP** | Window size | 8192 | 20% |
| | ACK number | 0 | 20% |
| | Checksum | 57876 | 7% |

Tabuľka 7.11: Zoznam položiek pre datovú sadu **ThorsHammer-DS4**.

Je však možné sledovať prítomnosť položiek **Header checksum**, **Widnow size** a **ACK number**, ktoré sú pre tento útok kľúčové a majú potenciál označiť veľkú časť útočiacich paketov. Výsledky aplikovania pravidiel zostavených z vybraných položiek je možné vidieť v nasledujúcej tabuľke:

| Data set | Rule | TP | FP |
|---|---|---|---|
| **TH-DS1** | Type of service = 0, Do not fragment = True<br>Window size = <5612, 11049><br>ACK number = 0, TCP Flags = SYN | 99.7% | 0% |
| **TH-DS2** | Type of service = 0<br>Total length = <0, 925><br>ACK number = <0, 1084045001> | 99.4% | 0.6% |
| **TH-DS3** | Type of service = 0<br>Total length = <0, 972><br>ACK number = <0, 1292624527> | 97.4% | 2.6% |
| **TH-DS4** | Header checksum = 0, Window size = 8192<br>ACK number = 0 | 54.3% | 3.3% |

Tabuľka 7.12: Výsledky aplikovania pravidiel na datové sady.

V prípade aplikácie pravidla na poslednú datovú sadu je možné vidieť značný percentuálny pokles hodnoty *True positive*. Tento pokles je dôsledkom zloženia datovej sady, kde je objem dát legitímnej komunikácie približne 4 násobne väščí ako pri datovej sade **ThorsHammer-DS2**. To ovplyvnilo hodnoty vybraných položiek, ktoré následne označili menší počet útočiacich paketov v datovej sade.

### 7.2.3 Hulk

Nakoľko bol pomocou tohto nástroja realizovaný útok na službu HTTP, bolo potrebné zmeniť aj referenčnú datovú sadu reprezentujúcu legitímnu komunikáciu. V tomto prípade bola legitímna komunikácia odfiltorvaná tak, aby obsahovala len HTTP pakety. Cieľom tohto experimentu bolo získať prehľad o tom, ako efektívne dokáže metóda pracovať nad dátami z aplikačnej vrstvy a zároveň využiť vybrané položky z týchto dát na označenie útočiacich paketov.

Nasledujúce tabuľky obsahujú hodnoty sledovaných položiek, ktoré tvoria profil legitímnej vzorky dát. Pre jednoduchosť budú uvedené hodnoty len pre vrstvy IP a TCP nakoľko vrámci HTTP protokolu je ukladaných niekoľko desiatok položiek z HTTP hlavičky.

V tabuľke 7.13 sa v niekoľkých bunkách vyskytuje symbol X, ktorý nahrádza hodnotu položky, ktorá nebola vypočítaná. V tomto prípade ide o najpočetnejšiu hodnotu položky a jej neuvedenie do profilu bolo zapríčinené tým, že sa v dátach vyskytovalo viac hodnôt s

| Type | Item | Mean | STD | Mode |
|---|---|---|---|---|
| **IP** | Header checksum | 4362 | 14677 | 0 |
| | Type of service | 0 | 0 | 0 |
| | Identification | 18683 | 8397 | X |
| | Fragment offset | 0 | 0 | 0 |
| | TTL | 128 | 0 | 128 |
| | Total length | 611 | 302 | 1001 |
| **TCP** | Data offset | 5 | 0 | 5 |
| | Window size | 7097 | 19364 | 256 |
| | Checksum | 36217 | 21458 | 59050 |
| | ACK number | 2258489705 | 1205570474 | X |
| | Sequence number | 2216662812 | 1220577846 | X |

Tabuľka 7.13: Štatistické hodnoty položiek v profile legitímnej komunikácie.

| Type | Item | Value |
|---|---|---|
| **IP** | More fragment | False = 100% |
| | Do not fragment | True = 100% |
| **TCP** | Urgent flag | False = 100% |
| | Flags | ACK = 3% |
| | | ACK + PSH = 97% |

Tabuľka 7.14: Percentuálne zastúpenie položiek profilu legitímnej komunikácie.

rovnakým počtom výskytov. Z toho dôvodu bol výpočet štatistík upravený tak, aby táto položka nebola zaradená do profilu a tým pádom nemohla zohrávať úlohu ani pri porovnávaní profilov.

Nasledujúce tabuľky obsahujú výsledky aplikácie navrhnutej metódy na datové sady, pričom je zaujímavé sledovať pokrytie dát položkami z HTTP hlavičky. V rámci generovania útoku boli niektoré hodnoty položiek HTTP hlavičky vyberané náhodným spôsobom z predom definovanej množiny hodnôt, iné mali definovanú statickú hodnotu.

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | TTL | 127 | 100% |
| **TCP** | Sequence number | <896127565, 3384971422> | 57% |
| | ACK number | <884494962, 3349342939> | 58% |
| **HTTP** | Origin | http://www.wis.fit.vutbr.cz | 100% |
| | Via | 1.0 fred, 1.1 example.com (Apache/1.1) | 100% |
| | Accept-encoding | identity | 99% |
| | Host | 192.168.100.10 | 100% |
| | Accept-charset | ISO-8859-1,utf-8;q=0.7,*;q=0.7 | 100% |
| | Connection | close | 100% |
| | Cache-control | no-cache | 100% |
| | Te | trailers | 100% |

Tabuľka 7.15: Zoznam položiek pre datovú sadu **Hulk-DS1**.

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | TTL | <109, 138> | 94% |
| | Total length | 1500 | 5% |
| **TCP** | Sequence number | <969254485, 3455317060> | 57% |
| | ACK number | <920406388, 3367674515> | 58% |
| **HTTP** | Origin | http://www.wis.fit.vutbr.cz | 99% |
| | Via | 1.0 fred, 1.1 example.com (Apache/1.1) | 100% |
| | Accept-encoding | identity | 68% |
| | Host | 192.168.100.10 | 67% |
| | Accept-charset | ISO-8859-1,utf-8;q=0.7,*;q=0.7 | 80% |
| | Connection | close | 69% |
| | Cache-control | no-cache | 99% |
| | Te | "trailers" | 100% |

Tabuľka 7.16: Zoznam položiek pre datovú sadu **Hulk-DS2**.

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | Total length | 1500 | 6% |
| **TCP** | Sequence number | <901353313, 3409026489> | 56% |
| | ACK number | <869366821, 3317630817> | 58% |
| **HTTP** | Accept-encoding | identity | 52% |
| | Host | 192.168.100.10 | 50% |
| | Accept-charset | ISO-8859-1,utf-8;q=0.7,*;q=0.7 | 65% |
| | Connection | close | 51% |

Tabuľka 7.17: Zoznam položiek pre datovú sadu **Hulk-DS3**.

V nasledujúcej tabuľke je zaujímavé sledovať vplyv datovej sady na vybrané položky. Keďže objem legitímnej komunikácie výrazne prevyšuje dáta útoku, je možné vidieť, že sa medzi tieto položky dostali už aj hodnoty, ktoré patria len do paketov legitímnej komunikácie. Tento fakt má samozrejme nepriaznivý dopad na výslednú efektívnosť zostaveného pravidla.

Opäť sa teda potvrdzuje slabé miesto navrhovanej metódy, kedy počet vybraných položiek, ktoré majú potenciál označiť útočiace pakety, silno závisí na kvalite útočiacej datovej sady. Práve tento fakt je jednou z hlavných príčin nemožnosti úplnej automatizácie celého procesu tvorby a aplikovania mitigačných pravidiel.

| Type | Item | Value | Coverage |
|---|---|---|---|
| **IP** | Total length | 1500 | 9% |
| **TCP** | Sequence number | <877972044, 3365146175> | 57% |
| | ACK number | <863430811, 3288580358> | 58% |
| **HTTP** | Accept-language | en-US,en;q=0.8 | 62% |
| | Accept-charset | ISO-8859-1,utf-8;q=0.7,*;q=0.3 | 56% |
| | Accept | */* | 69% |

Tabuľka 7.18: Zoznam položiek pre datovú sadu **Hulk-DS4**.

Nasleduje prehľad dosiahnutých výsledkov po aplikácií vybraných položiek pre jednotlivé datové sady. Prínos analyzovania dát z aplikačnej vrstvy je možné sledovať pri datovej sade **Hulk-DS3**, pre ktorú boli vytvorené dve mitigačné pravidlá. Prvé z nich obsahuje len položky z HTTP hlavičky, vďaka ktorým je úspešnosť označenia útočiacich paketov až 99.9%. Oproti tomu druhé pravidlo zostavené len z položky **ACK number** jasne ukazuje slabé pokrytie útočiacich paketov a zároveň vysokú mieru falošne poztívnych označení. V prípade poslednej datovej sady, bolo už z hodnôt uvedených v tabuľke 7.18 zrejmé, že efektívnosť zostaveného pravidla bude minimálna.

| **Data set** | **Rule** | **TP** | **FP** |
|---|---|---|---|
| **Hulk-DS1** | Accept-charset = 'ISO-8859-1,utf-8;q=0.7,*;q=0.7'<br>Via = '1.0 fred, 1.1 example.com (Apache/1.1)'<br>Origin = 'http://www.wis.fit.vutbr.cz'<br>Accept-encoding = 'identity'<br>Host = '192.168.100.10', Te = 'trailers'<br>Connection = 'close', Cache-control = 'no-cache'<br>TTL = 127 | 99.9% | 0% |
| **Hulk-DS2** | Accept-charset = 'ISO-8859-1,utf-8;q=0.7,*;q=0.7'<br>Via = '1.0 fred, 1.1 example.com (Apache/1.1)'<br>Origin = 'http://www.wis.fit.vutbr.cz'<br>Accept-encoding = 'identity', Te = 'trailers'<br>Connection = 'close', Cache-control = 'no-cache'<br>TTL = <109, 138> | 100% | 0% |
| **Hulk-DS3** | Accept-charset = 'ISO-8859-1,utf-8;q=0.7,*;q=0.7'<br>Accept-encoding = 'identity', Connection = 'close'<br>Host = '192.168.100.10' | 99.9% | 0% |
| **Hulk-DS3** | ACK number = <869366821, 3317630817> | 49.7% | 50.3% |
| **Hulk-DS4** | ACK number = <863430811, 3288580358><br>Accept-language = 'en-US,en;q=0.8'<br>Accept = '*/*' | 0% | 100% |

Tabuľka 7.19: Výsledky aplikovania pravidiel na datové sady.

## 7.3 Iné možnosti pri analýze dát

Existuje niekoľko nástrojov, ktoré využívajú rôzne prístupy pri spracovaní dát a mohli by byť alternatívou k metóde popisovanej v tejto práci. Jedným z týchto nástrojov je aj Weka[1], ktorá je určená pre prácu s dátami na rôznej úrovni, či už ide o ich klasifikáciu, vizualizáciu alebo triedenie. Ponúka niekoľko algoritmov pomocou ktorých je možné realizovať klasifikáciu vstupných dát a práve z toho dôvodu bola vybraná ako vhodný kandidát pre porovnanie s navrhnutou metódou.

Pri návrhu metódy bola zvažovaná aj možnosť klasifikácie útočiacich dát, no pre možné problémy s dĺžkou trvania tohto procesu by zrejme nebolo možné túto metódu použiť v reálnom čase. Navyše by bolo potrebné vyriešiť ďalšie problémy spojené so samotnou klasifikáciou ako napríklad správny výber testovacej množiny dát, ktorá by sa použila vo fáze učenia. Keďže tento nástroj ponúka aj širokú škálu vizualizačných pomôcok, je možné

[1] http://www.cs.waikato.ac.nz/ml/weka/

pomocou nich získať odlišný pohľad na analyzované dáta, čo môže napomôcť k ďalšiemu rozvoju navrhnutej metódy.

Nad niektorými datovými sadami boli realizované pokusy o klasifikáciu pomocou tohto nástroja. Keďže sa v tomto prípade jedná o strojové učenie s učiteľom, bolo potrebné vytvoriť tréningovú datovú sadu, ktorá pozostávala z časti datovej sady obsahujúcej legitímnu komunikáciu a z časti útočiacich dát. Ako referenčná datová sada reprezentujúca útok, bola zvolená datová sada **ThorsHammer-DS1**, ktorá obsahovala len dáta útoku.

Metóda pracuje spôsobom kedy si na základe tréningovej sady zostaví rozhodovací strom na základe ktorého je neskôr vykonávaná klasifikácia vstupných dát. Výsledkom sú hodnoty *True positive* a *False positive* na základe ktorých je možné zistiť do akej miery bola klasifikácia úspešná. Hodnota *True positive* bola pre túto datovú sadu po vykonaní klasifikácie na úrovni 61.8% a hodnota *False positive* dosahovala 38.2%.

Ako je možné vidieť miera falošne pozitívnych označení je už pri datovej sade obsahujúcej len dáta útoku neprimerane vysoká. Po bližšej analýze bolo zistené, že rozhodovcí strom podľa ktorého sa klasifikácia vykonávala obsahoval len niekoľko podmienok, v ktorých hlavným prvkom boli položky **Identification** z IP vrstvy a **TCP Flags**. Rozhodovanie na základe položky **Identification** prispelo k zvýšeniu hodnoty falošne pozitívnych označení. Naopak v rozhodovacom strome absentovali položky ako **Type of service** alebo **Window size**, ktoré majú potenciál označiť veľkú časť útočiacich paketov. Pre porovnanie je zoznam položiek, ktorý bol výstupom popisovanej metódy, zobrazený v tabuľke 7.8. Experimenty vykonané nad ostatnými datovými sadami, nepriniesli výrazné zlepšenie výsledkov oproti prezentovaným hodnotám.

## 7.4 Zhodnotenie výsledkov metódy

Z prezentovaných výsledkov v predchádzajúcich častiach je možné vidieť istú závislosť medzi kvalitou datovej sady a efektivitou vytvoreného mitigačného pravidla. V prípade, že v útočiacej datovej sade je objem legitímnej komunikácie menší ako objem dát útoku, je vysoko pravdepodobné, že navrhnutá metóda vyberie položky, ktoré majú dostatočný potenciál označiť veľkú časť útočiacich paketov.

Pri tomto zložení datovej sady by bolo možné uvažovať aj o automatizovaní metódy, kedy by sa z vybraných položiek a základe dosiahnutej hodnoty percentuálneho pokrytia dalo zostaviť mitigačné pravidlo, ktoré by bolo dostatočne účinné a zároveň by jeho aplikáciou nedošlo k obmedzeniu legitímnej komunikácie na cieľovú stanicu. Bolo by však nutné stanoviť hraničnú hodnotu celkového pokrytia, na základe ktorej by sa určilo či vybraná položka bude súčasťou pravidla alebo nie.

Tento návrh by však s najväčšou pravdepodobnosťou nebolo možné realizovať v prípadoch, kedy by zloženie útočiacej datovej sady bolo podobné datovým sadám s označením **DS2-DS4**. Z pohľadu označnia kľúčových položiek výsledky experimentov pre tieto datové sady dopadli v rámci očakávaní.

Problémom vybraných položiek z týchto datových sád je ich celkové pokrytie, na základe ktorého by nebolo možné automatizovaným spôsobom jednoznačne vybrať položky pre mitigačné pravidlo. V niektorých experimentoch bolo jasne vidieť, že aj napriek tomu že vybraná položka mala hodnotu pokrytia výrazne menšiu než ostatné, stále mala potenciál označiť veľkú časť útočiacich paketov. Z toho dôvodu je v týchto prípadoch stále potrebná určitá miera ručnej analýzy výsledkov.

Navrhnutá metóda však potvrdila stanovenú hypotézu v dobe návrhu, ktorá predpokladala, že v prípade špecifických útokov bude možné pomocou nej označovať pakety týchto

útokov. Je jasné, že v prípade dostatočnej znalosti danej problematiky je možné vytvoriť útok, ktorého charakteristika bude len veľmi ťažko odlíšitelná od bežnej komunikácie, avšak pre útoky vytvorené pomocou bežne dostupných nástrojov je možné na základe tejto metódy dosiahnuť relatívne uspokojivú mieru označení útočiacich paketov.

# Kapitola 8

# Záver

V prvej kapitole boli popísane základné princípy monitorovania sietí, ich rozdelenie a kľučové rozdiely medzi jednotlivými prístupmi. Boli objasnené pojmy ako Netflow, IPFIX, sFlow, IP tok a iné. Následne bol opísaný princíp a architektúra protokolu NetFlow, o ktorý sa prostredníctvom vyvíjaného modulu táto práca opiera.

Nasledujúca kapitola objasňovala postupy akými je možné detegovať bezpečnostné hrozby na sieti. V tejto časti boli prezentované systémy IDS, ich funkcia a princíp činnosti ako aj základné rozdelenie. Podrobne boli prezentované dva hlavné prístupy, na ktorých sú tieto systémy založené a ich zástupcovia. Ďalej bol predstavený framework NEMEA, ktorý tvorí základ pre vývoj navrhovaného modulu. Boli popísané jeho základné stavebné prvky, spôsob komunikácie medzi nimi a bol stručne predstavený formát posielaných správ. Taktiež boli objasnené pojmy súvisiace s týmto frameworkom ako napríklad TRAP a UniRec. Záver kapitoly sa venoval systému spätného záchytu dát, ktorý slúži na zachytávanie kompletnej komunikácie v prípade, že táto komunikácia je označená za podozrivú.

V kapitole 4 boli uvedené nástroje, ktoré slúžia na analýzu dát, z ktorých sa získavajú informácie potrebné na vytváranie profilov popisujúcich rôzne druhy útokov. Následne boli rozobraté vybrané druhy bezpečnostných hrozieb a ich základné charakteristiky. Pri každej hrozbe boli uvedené aj charakteristické rysy, ktoré by sa dali použiť na ich detekciu. V nasledujúcej kapitole bol prezentovaný návrh tvorby mitigačných pravidel a jednotlivé položky z ktorých sa mitigačné pravidlo skladá. Kapitola 6 obsahovala popis implementácie celej metódy. Bol podrobne rozobratý proces získavania štatistík z vybraných položiek ako aj kompletný postup pri porovnávaní zostavených profilov komunikácie. Záver kapitoly sa venoval systému overovania pokrytia útočiacej datovej sady.

Záverečná kapitola 7 obsahuje popis realizovaných experimentov nad datovými sadami získanými pomocou zvolených nástrojov pre generovanie DDoS útokov. Z dosiahnutých výsledkov je možné sledovať určitú závislosť medzi zložením datovej sady reprezentujúcej útok a efektivitou navrhovanej metódy. Ďalej sa osvedčilo vyhodnocovanie dát z aplikačnej vrsvtvy, pomocou ktorého bolo možné jednoznačne identifikovať útočiace pakety aj v prípade väčšieho množstva legitímnej komunikácie v datovej sade. Výsledky experimentov ďalej naznačili, že tvorbu mitigačných pravidiel nieje možné úplne automatizovať, nakoľko výber správnych položiek nieje triviálny a závisí od viacerých faktorov. Toto zistenie vytvára priestor pre ďalší rozovoj metódy, ktorý by mohol smerovať k snahe definovať postup, ktorý by rozhodol o tom, či je z vybraných položiek možné automaticky zostaviť pravidlo, čo by v niektorých prípadoch umožnilo automatizovanie celého procesu.

# Literatúra


[1] Bro Documentation - Introduction [online]. December 2016 [cit. 2016-12-09].
URL http://www.bro.org/sphinx/intro/#introduction

[2] NetFlow. PandoraFMS [online]. Máj 2012 [cit. 2015-04-27].
URL http://wiki.pandorafms.com/index.php?title=Pandora:Documentation_en:Netflow

[3] Introduction to Cisco IOS NetFlow - A Technical Overview [online]. Máj 2012 [cit. 2016-11-04].
URL http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html

[4] Cejka, T.; Bartos, V.; Svepes, M.; aj.: NEMEA: A Framework for Network Traffic Analysis. In *12th International Conference on Network and Service Management (CNSM 2016)*, 2016.

[5] Cisco Systems: *NetFlow Services Solutions Guide [online]*. Júl 2001 [cit. 2016-11-8].
URL http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/netflow/nfwhite.html#wp1032617

[6] Cisco Systems: *NetFlow Reliable Export With SCTP [online]*. April 2012 [cit. 2016-11-7].
URL http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/netflow/configuration/12-4t/nf-12-4t-book/nflow-export-sctp.html

[7] Cisco Systems: *Cisco NetFlow Collector User Guide: Overview [online]*. Február 2014 [cit. 2016-11-7].
URL http://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/6-0/tier_one/user/guide/user/overview.html

[8] Cisco Systems: *NetFlow Configuration Guide, Cisco IOS Release 12.4T [online]*. Apríl 2012 [cit. 2016-11-08], str. 44-45.
URL http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/netflow/configuration/12-4t/nf-12-4t-book.pdf

[9] Cisco Systems: *NetFlow Version 9 Flow-Record Format [online]*. Máj 2011 [cit. 2016-11-08].
URL http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html

[10] Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954, Október 2004.
URL http://www.rfc-editor.org/rfc/rfc3954.txt

[11] Claise, B.; Trammell, B.; Aitken, P.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. STD 77, September 2013.
URL http://www.rfc-editor.org/rfc/rfc7011.txt

[12] Eddy, W.: TCP SYN Flooding Attacks and Common Mitigations. RFC 4987, August 2007.
URL https://tools.ietf.org/html/rfc4987

[13] Kurose, J. F.; Ross, K. W.: *Computer Networking: A Top-Down Approach (6th Edition)*. Pearson, 6 vydání, 2012, ISBN 0132856204, 9780132856201.

[14] Petr Matoušek: *Síťové aplikace a jejich arichtektura.* Brno: VUTIUM, 2014, ISBN 9788021437661.

[15] Phaal, P.; Panchen, S.; McKee, N.: InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. RFC 3176, September 2001.
URL http://www.rfc-editor.org/rfc/rfc3176.txt

[16] Postel, J.: Transmission Control Protocol. STD 7, September 1981.
URL http://www.rfc-editor.org/rfc/rfc793.txt

[17] Roesch, M.: Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th USENIX Conference on System Administration*, LISA '99, Berkeley, CA, USA: USENIX Association, 1999, s. 229–238.

[18] Rosa, Z.; Cejka, T.; Zadnik, M.; aj.: Building a Feedback Loop to Capture Evidence of Network Incidents. In *12th International Conference on Network and Service Management (CNSM 2016)*, 2016.

[19] sFlow.org: *Traffic Monitoring using sFlow [online].* 2013 [cit. 2016-11-14], str. 1-4.
URL http://www.sflow.org/sFlowOverview.pdf

[20] Stewart, R.: Stream Control Transmission Protocol. RFC 4960, September 2007.
URL http://www.rfc-editor.org/rfc/rfc4960.txt

[21] Trost, R.: *Practical Intrusion Analysis - Prevention and Detection for the Twenty-First Century.* Addison-Wesley Professional, první vydání, 2009, ISBN 9780321591807.