VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

ANALÝZA ZPĚTNĚ ROZPTÝLENÉHO DDOS PROVOZU V DATECH O SÍŤOVÝCH TOCÍCH

ANALYSIS OF DDOS BACKSCATTER TRAFFIC IN NETWORK FLOW DATA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. MARTIN MARUŠIAK**
AUTHOR

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

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2020/2021

Zadání diplomové práce

Student: **Marušiak Martin, Bc.**
Program: Informační technologie
Obor: Bioinformatika a biocomputing
Název: **Analýza zpětně rozptýleného DDoS provozu v datech o síťových tocích**
Analysis of DDoS Backscatter Traffic in Network Flow Data
Kategorie: Počítačové sítě
Zadání:

1. Seznamte se s měřením síťového provozu na základě toku a dále se seznamte s problematikou útoků na odepření služby, DDoS (Distributed Denial of Service).
2. Nastudujte relevantní literaturu k problematice detekce zpětně rozptýleného DDoS provozu a diskutujte zejména použití dat z teleskopů a honeypotů.
3. Navrhněte přístup pro detekci zpětně rozptýleného provozu na základě dat o síťových tocích.
4. Implementujte navržený přístup formou prototypu.
5. Ověřte navržený přístup na dostupných datových sadách včetně reálného provozu a diskutujte možnosti dalšího pokračování.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Žádník Martin, Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 19. května 2021
Datum schválení: 30. října 2020

Abstrakt

Táto práca sa zaoberá problematikou detekcie útokov odopretia služby (DoS) využívajúcich metódu náhodného podvrhnutia zdrojovej IP adresy v útočných paketoch. Tento typ DoS útokov po sebe zanecháva stopu v podobe tzv. spätného rozptylu, na základe ktorého je možné identifikovať obeť útoku. Analýza spätného rozptylu a jeho použitie k detekcii DoS útokov bola doposiaľ limitovaná na nevyužité adresové rozsahy označované ako sieťové teleskopy. V rámci tejto práce bola preto navrhnutá metóda, ktorá dokáže detegovať DoS útoky zo spätného rozptylu aj mimo prostredia teleskopov za prítomnosti legitímnej prevádzky a to na navyše z dát sieťových tokov. Navrhnutá metóda bola implementovaná v rámci systému NEMEA a vyhodnotená na reálnych dátach tokov poskytnutých organizáciou CESNET.

Abstract

This work focuses on detection of denial of service (DoS) attacks which utilize random spoofing of source IP address in attack packets. These types of attacks lead to generation of side effect in a form of backscatter that can be used to identify victims of such attacks. Backscatter analysis has so far been limited to unused address space ranges referred to as network telescopes. This work therefore proposes a new method of DoS attack detection via backscatter outside of network telescope environment where legitimate user traffic is also present. Furthermore proposed approach uses only abstracted traffic in a form of network flows. Presented method was implemented as part of NEMEA system and tested on real flow data capture provided by CESNET.

Kľúčové slová

DoS, DDoS, NetFlow, sieťové toky, sieťový teleskop, spätný rozptyl, strojové učenie

Keywords

DoS, DDoS, NetFlow, network flow, network telescope, backscatter, machine learning

Citácia

MARUŠIAK, Martin. *Analýza zpětně rozptýleného DDoS provozu v datech o síťových tocích.* Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Žádník, Ph.D.

Analýza zpětně rozptýleného DDoS provozu v datech o síťových tocích

Prehlásenie

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

.....
Martin Marušiak
17. mája 2021

Poďakovanie

Chcem sa poďakovať vedúcemu práce Ing. Martinovi Žádníkovi, Ph. D. za odborné vedenie práce a konzultácie k danej problematike. Ďalej sa chcem poďakovať organizácii CESNET za poskytnutie dát tokov, organizácii CAIDA za udelenie prístupu k dátam teleskopu a organizácii MetaCentrum za poskytnutie výpočtových zdrojov v rámci projektu CERIT Scientific Cloud (LM2015085) a projektu CESNET (LM2015042) financovaných z programu MŠMT Projekty veľkých infraštruktúr pre VaVaI.

Obsah

Kapitola 1

Úvod

Útoky odopretia služby (DoS) predstavujú kategóriu útokov, ktorých cieľom je znemožniť legitímnym užívateľom prístup k danej službe. Často vyskytujúce sa varianty DoS útokov sú principiálne pomerne jednoduché a k uskutočneniu používajú hlavne metódu zahltenia cieľovej služby veľkým množstvom požiadaviek. Napadnutá služba potom pod týmto náporom nedokáže spracovať požiadavky oprávnených užívateľov vôbec alebo dôjde k značnému spomaleniu danej služby. Na útoku sa pritom typicky účastní niekoľko zariadený a v tom prípade hovoríme o distribuovanej variante DoS útokov (DDoS). Nakoľko je DDoS útok bežnou variantov DoS útoku sú tieto pojmy v literatúre často navzájom zamieňané a preto budú v tejto práci vo väčšine prípadov považované oba výrazy za ekvivalentné.

Typickými cieľmi DoS útokov sú webové stránky, databázové aplikácie či online hry. DoS útoky dokážu napáchať nemalé finančné škody vo forme odlivu zákazníkov a straty reputácie. Finančné prostriedky vynaložené na preventívnu obranu proti týmto útokom tiež nie sú zanedbateľné. Relevancia DoS útokov ako internetovej hrozby stále rastie, a to aj vďaka dostupnosti tohto typu útoku pre širokú verejnosť vo forme rôznych nástrojov a platených služieb ponúkajúcich uskutočnenie týchto útokov bez nutnosti ich odbornej znalosti. S cieľom utajenia identity útočník navyše často pozmení svoju IP adresu tak, aby bolo znemožnené jeho vypátranie. Za týmto účelom je bežne používaná najmä metóda náhodného podvrhnutia zdrojovej adresy v útočnom pakete, ktorá zároveň sťažuje proces detekcie a mitigácie DoS útoku. Dôsledkom tohto podvrhnutia je vedľajší jav, kedy napadnutá služba zasiela odpovede na podvrhnuté adresy. Nakoľko sú adresy podvrhnuté náhodne, môže takto vytvorená odpoveď doraziť na ktorúkoľvek smerovateľnú IP adresu na internete. Tieto pakety sa v literatúre súhrne označujú pojmom spätný rozptyl (*backscatter*). S použitím spätného rozptylu je potom možné detegovať prebiehajúci DoS útok aj mimo siete zariadenia, na ktoré je útok vedený. Na účely detekcie DoS útokov zo spätného rozptylu je potrebné mať k dispozícii pomerne veľký adresový priestor a čím väčší je tento priestor, tým viac je pravdepodobné, že pakety s náhodne podvrhnutou adresou dorazia práve sem. Analýza a detekcia DoS útokov sa preto skúma na tzv. teleskopoch. Teleskopy sú veľké nepriradené monitorované adresové bloky charakteristické absenciou legitímnej komunikácie. Medzi prevádzku, ktorá na teleskopy smeruje, patria scany, spätný rozptyl a pakety doručené na základe chybnej sieťovej konfigurácie. Vzhľadom na absenciu legitímnych zariadení je detekcia DoS útokov v prostredí teleskopu pomerne priamočiara a dokáže si vystačiť len s kvantitatívnymi prahmi v kombinácii s jednoduchými podmienkami na príznaky v hlavičke paketu. Vytvorenie vlastného teleskopu je však dnes kvôli nedostupnosti IPv4 adries pomerne problematické. Táto práca sa preto venuje možnostiam identifikovania

DoS útokov zo spätného rozptylu v bežných sieťach, kde sú prítomné legitimné zariadenia a to navyše len s použitím dát vo forme sieťových tokov.

V kapitole 2 je popísaná práca so sieťovými dátami vo forme tokov spolu s uvedením architektúry NetFlow. Táto kapitola sa ďalej venuje výhodám a nevýhodám reprezentácie v podobe sieťových tokov z hľadiska efektivity a neskoršej analýzy týchto dát. Následne sú v kapitole 3 rozobrané rôzne typy DoS útokov a ich vzťah k mechanizmu podvrhnutia IP adries v útočnom pakete. Kapitola 4 potom ponúka štatistický pohľad na možnosť detekcie DoS útokov skrz spätný rozptyl a kapitola 5 prezentuje niekoľko existujúcich prístupov využitia tohto javu k detekcii DoS útokov na teleskopoch. Nasledujúce tri kapitoly sa venujú metóde detekcie DoS útokov vytvorenej v rámci tejto práce. Jedná sa o kapitolu 6, kde je metóda navrhnutá, kapitolu 7, ktorá sa zaoberá jej implementáciou v rámci systému NEMEA a kapitolu 8 vyhodnocujúcej implementovaný prístup na dátach tokov organizácie CESNET.

Kapitola 2

Meranie sieťovej komunikácie pomocou tokov

Meranie sieťovej komunikácie a jej monitorovanie je možné rozdeliť na dva prístupy: pasívny a aktívny [14]. Pri pasívnom prístupe dochádza k sledovaniu a ukladaniu prebiehajúcej komunikácie, z ktorej sú čerpané informácie o stave siete. Jedná sa napríklad o použitie logovacích údajov zbieraných pomocou protokolu syslog. Do tejto kategórie taktiež patrí zachytávanie paketov na sieti a ich abstrahované ukladanie v podobe tokov, kde je typickým predstaviteľom architektúra NetFlow od firmy Cisco.

V prípade aktívneho monitorovania administrátor resp. aplikácia aktívne overuje stav sieťových zariadení a služieb. Jednoduché aktívne overovanie je možné uskutočniť napríklad pomocou protokolov ICMP a telnet. Ďalej tu patria nástroje na zistenie dostupnosti zariadenia ping či trasy traceroute. Pokročilejší príklad aktívneho monitorovania predstavuje architektúra SNMP a jej neskoršie rozšírenie v podobe RMON.

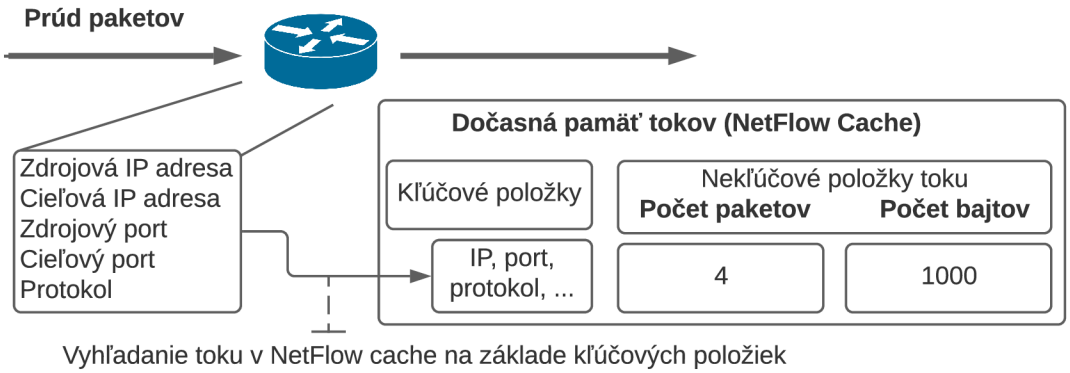
Pasívny pristúp detekcie pomocou tokov, NetFlow, tvorí hlavný obsah tejto kapitoly. Pojem NetFlow má v literatúre viacero významov, môže sa ním myslieť už spomínaná architektúra NetFlow od firmy Cisco alebo protokoly použité pre prenos dát tokov z meracieho bodu do zbernej stanice (kolektoru). Označenie NetFlow je však rozšírené do takej miery, že sa bežne stotožňuje s technológiou extrakcie a zberu tokov ako takou [10]. V tejto kapitole bude pojem NetFlow používaný v jeho obecnom význame.

2.1 NetFlow

Základným konceptom NetFlow je tok (*flow*). Tok predstavuje množinu paketov so spoločnými vlastnosťami, ktoré prechádzajú meracím bodom siete počas určitého časového intervalu [35]. Spoločné vlastnosti predstavujú typicky prvky hlavičky paketu, ako porty a IP adresy. Niekoľkým paketom so spoločnými vlastnosťami teda odpovedá jeden tok. Toky navyše neobsahujú samotný obsah agregovaných paketov (*payload*), čo spôsobuje výraznú redukciu objemu dát potrebných na uloženie toku. Monitorovanie pomocou tokov je preto možné aj na vysokorýchlostných sieťach, ktorými prechádza veľké množstvo komunikácie. Nasadenie technológie NetFlow je vďaka týmto vlastnostiam pomerne rozšírené a prácu s tokmi podporuje mnoho sieťových prvkov, ako smerovače, prepínače a bezpečnostné brány (*firewall*) [10].

Tok má dve časti: kľúčovú (*key*) a nekľúčovú (*non-key*) [10]. Kľúčová časť definuje položky, na základe ktorých sa posudzuje príslušnosť paketu k toku. Ku kľúčovým položkám

typicky patrí zdrojová IP adresa, cieľová IP adresa, cieľový port, zdrojový port a protokol. Dôsledkom predošlej voľby kľúčových položiek je jednosmernosť tokov. To znamená, že pre dva vzájomne komunikujúce zariadenia bude vytvorený osobitný tok v každom smere. Nekľúčové položky toku predstavujú hodnotu odvodenú z množiny paketov daného toku, ako napríklad: počet paketov v toku, suma veľkostí obsahu paketov, časová značku začiatku (prvý paket) a konca toku (posledný paket). Všetky tieto prvky sú znázornená na obrázku 2.1. Vidíme, že sieťové zariadenie postupne na základe kľúčových položiek spája prichádzajúce pakety do tokov a dopočítava nekľúčové položky. Toky sú v rámci zariadenia uchovávané v dočasnej pamäti označovanej ako tzv. *NetFlow cache* odkiaľ budú neskôr odoslané na kolektor.



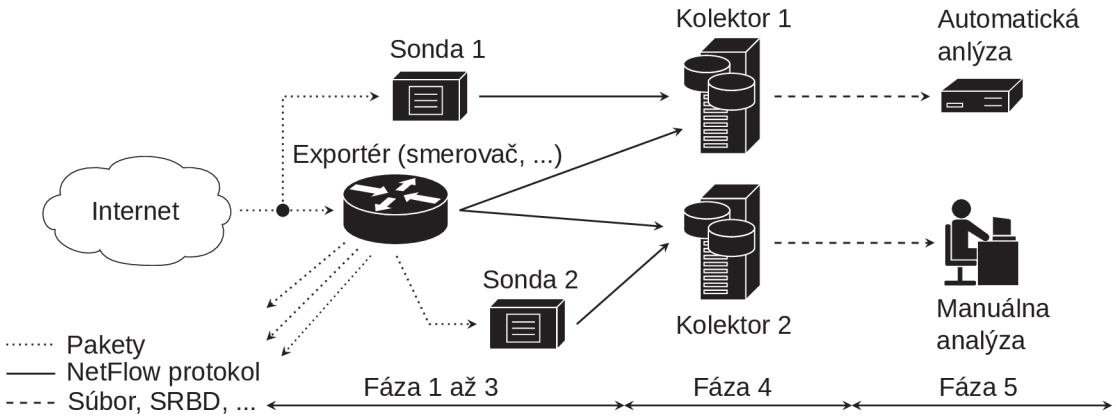
Obr. 2.1: Agregácia paketov so spoločnými vlastnosťami na meracom zariadení do toku.

2.2 Architektúra NetFlow

Monitorovanie siete pomocou tokov sa skladá z niekoľkých fáz [10]:

1. Zber paketov a ich predspracovanie (*packet observation*)
2. Tvorba tokov (*metering*)
3. Export tokov (*exporting*)
4. Kolekcia dát (*data collection*)
5. Analýza dát (*data analysis*)

Fázy 1 až 3 bývajú často vykonané v rámci jedného zariadenia označovaného ako exportér (*flow export device*). Exportérom môže byť napríklad smerovač či prepínač, ak je exportérom zariadenie špecificky vyhradené za účelom monitorovania tokov, potom sa tiež označuje pojmom sonda (*flow probe*). Dáta tokov sa v rámci fázy 4 zasielajú NetFlow protokolom z exportérov, resp. sond na kolektor. Takto zozbierané dáta tokov sú následne analyzované automatizovane alebo za účasti administrátora v piatej fáze. Pri analýze tokov dochádza typicky k tvorbe súhrnných štatistík o stave siete, profilovaniu komunikácie a detekcii anomálií. Uvedené fázy a prvky architektúry NetFlow sú znázornené na obrázku 2.2.



Obr. 2.2: Prvky a fázy v architektúre NetFlow [10].

Exportér

Ako bolo uvedené vyššie, exportér reprezentuje až tri fázy: zber paketov, tvorbu tokov a ich export [10]. V prvej fáze dochádza k zachyteniu resp. príchodu paketu na sieťové zariadenie, ktorému je následne priradená časová značka. Zachytený paket je ďalej orezaný na určitú veľkosť (*snapshot length*) nakoľko obsah paketu mimo hlavičiek nie je zvyčajne k tvorbe toku potrebný. Voliteľne je možné nastaviť vzorkovanie a filtrovanie paketov, z ktorých budú následne zostavené toky. Vzorkovanie je založené na náhodnom či deterministickom výbere paketov v závislosti od poradia ich príchodu a filtrovanie sa vzťahuje k určitej vlastnosti paketu. Vzorkovanie je používané za účelom zníženia počtu paketov určených na spracovanie a filtrovanie limituje pakety na skúmanú doménu, napríklad špecifický port či IP adresu. V oboch prípadoch však dochádza k zmenšeniu objemu dát, ktoré je potrebné v ďalších krokoch spracovávať.

Toky sú na exportéri uložené v dočasnej pamäti. Príchodom paketu do druhej fázy spracovania sa v tejto pamäti vytvorí nový tok, ak ešte neexistuje položka s odpovedajúcou n-ticou kľúčových hodnôt. V prípade výskytu kľúčovej n-tice v dočasnej pamäti dôjde len k aktualizovaniu nekľúčových hodnôt príslušného toku. Tento proces je znázornený na obrázku 2.1. Obsah dočasnej pamäte je pravidelne exportovaný pomocou NetFlow protokolu na kolektor. Pri exporte toku dochádza k jeho uvoľneniu z dočasnej pamäte, čo znamená, že dôjde k vymazaniu informácií, ktoré boli s daným tokom asociované. Fáza exportu môže nastať v rôznych prípadoch:

- Aktívny časovač – predstavuje maximálnu dobu zotrvania toku v dočasnej pamäti. Typické hodnoty sú v rozsahu 120 sekúnd až 30 minút. Aktívny časovač je užitočný pri monitorovaní dlhotrvajúcej komunikácie, tú by bolo bez použitia tohto časovača možné pozorovať na kolektore až po jej skončení.

- Neaktívny časovač – stanovuje limit na čas uplynutý od posledného príchodu paketu daného toku. Typické hodnoty sú v rozmedzí 15 sekúnd až 30 minút. Ak je tento limit prekročený, dôjde k exportu toku. Dosiahnutie limitu indikuje ukončenie komunikácie a preto nemá význam tok ďalej držať v pamäti.

- Prirodzené uvoľnenie – tok je uvoľnení na základe znalosti protokolu. Jedná sa napríklad o detekciu RST či FIN príznakov pri protokole TCP, ktoré naznačujú koniec komunikácie.

- Zaplnenie dočasnej pamäte tokov – je potrebné uvoľniť pamäť pre dáta ďalších tokov.

Pred exportom môže navyše podobne ako pri vzorkovaní a filtrovaní paketov dôjsť k vzorkovaniu filtrovaniu na úrovni tokov [10].

Kolektor

Kolektor môže prijímať NetFlow dáta z jedného alebo viacerých exportérov. Okrem zhromažďovania dát plní kolektor aj mnohé ďalšie funkcie, ako ich kompresiu, agregáciu, anonymizáciu, filtrovanie a tvorbu súhrnných prehľadov o komunikácii [10].

Hoci použitie tokov pomerne výrazne redukuje výpočtové a priestorové nároky v porovnaní s paketovým spracovaním stále sa môže jednať o terabajty dát, ktoré je nutné spracovať a uchovať [10]. Dáta sieťových tokov sú typicky komprimovateľnej povahy a použitie kompresných metód je preto žiadúce [33]. Reprezentácia komunikácie v podobe tokov spolu s kompresiou vedie k viac ako 2 000-násobnej úspore priestoru oproti pôvodným paketovým dátam a to aj bez uvažovania vzorkovania či filtrovania na exportéri [10].

Nároky na úložný priestor je ďalej možné redukovať agregáciou v závislosti od požiadaviek a cieľov, ktoré sú na systém kladené [34]. Napríklad pre dlhodobé účely z hľadiska pozorovania trendov vyťaženia sieťových liniek je vhodné použiť vyššiu mieru agregácie.

Toky môžu byť na kolektore uložené v rôznej forme [10]:

- Ploché súbory (*flat files*) – reprezentujú dáta tokov vo forme binárnych a textových súborov. Nové dáta sú pridávané na koniec týchto súborov, čo ich robí z hľadiska operácie zápisu pomerne efektívnymi. Priestor potrebný na ich uloženie je typicky menší ako u databázových súborov pretože nevyužívajú indexy. Takto uložené dáta sú však viac limitované z pohľadu možnosti tvoriť pokročilejšie dotazy. Z neexistencie indexu vyplýva nutnosť prehľadania celého súboru pri dotazovaní. To ale v skutočnosti nie je až tak veľký problém, nakoľko dáta tokov väčšinou nie sú uložené v jednom plochom súbore, ale vo viacerých menších súboroch. Dáta v menších súboroch sa typicky viažu k určitému časovému intervalu. Toto usporiadanie potom predstavuje časový kvázi index a teda nie je nutné prehľadávať všetky dáta. Tento spôsob organizovania dát využíva napríklad nástroj `nfdump`.
- Relačné (riadkovo orientované) databáze – jedná sa o uloženie pomocou klasických databázových systémov (SRBD) ako `MySQL`, `PostgreSQL` či `Miscrosoft SQL Server`.
- Stĺpcové databáze – dáta sú orientované po stĺpcoch (atribúty tokov). Vykonanie dotazu zahŕňa len prístup k zvoleným typom atribútov narozdiel od riadkových databáz, kde sa do pamäte môžu načítavať aj hodnoty atribútov, ktoré nie sú nutné na zodpovedanie dotazu. Orientácia po stĺpcoch ďalej umožňuje vyššiu mieru kompresie pretože hodnoty v rámci jedného atribútu majú typicky homogénny charakter. Predstaviteľom je `FastBit`.

Pri uchovaní dát je taktiež potrebné dbať na ochranu súkromia. Z hľadiska anonymizácie sú na tom toky výrazne lepšie ako paketový záchyt už len preto, že neuchovávajú obsah paketov. Z dát tokov je ale stále možné odvodiť informáciu o tom aké zariadenia spolu komunikujú. Pri úvahách o bezpečnosti sa vychádza zo zjednodušeného predpokladu, že IP adresa identifikuje užívateľa. Preto je potrebné riešiť to, ako by sa mali tieto informácie ukladať, čo je predmetom legislatívy daného štátu. Je dobré si však uvedomiť, že anonymizácia úzko súvisí s cieľmi analýzy. Napríklad pre celkové štatistiky prenesených dát za určité

obdobie nie je nutné mať k dispozícii informáciu o IP adresách, ak však ale bude úlohou určiť vzťahy medzi jednotlivými časťami monitorovacej siete, potom môže anonymizácia analýzu skomplikovať [10].

Komunikačné protokoly pre výmenu NetFlow dát

V tejto časti sú popísané tri najpoužívanejšie protokoly na prenos tokov z exportéru na kolektor: NetFlow v5, NetFlow v9 a IPFIX. Protokoly NetFlow v5 a 9 pochádzajú od firmy Cisco, kde verzia 5 bola prvou verziou, ktorá sa rozšírila vo výraznejšom merítku a 9 predstavuje jej neskoršie vylepšenie. Paralelne s protokolom NetFlow v9 organizácia IETF vyvinula a štandardizovala protokol IPFIX [10].

Prenos spojený s výmenou NetFlow dát medzi exportérmi a kolektorom je pomerne efektívny a predstavuje veľmi malú časť z celkovej komunikácie na sieti. Jedná sa typicky len o zlomky percenta s hodnotami zastúpenia dát tokov v sieťovej komunikácii v rozmedzí od 0,1 do 0,4 % a to bez použitia vzorkovania či filtrovania [10, 37, 22].

NetFlow v5

Stále sa jedná o najviac rozšírenú verziu NetFlow protokolu podporovanú mnohými zariadeniami [27]. Najväčšia nevýhoda tohto protokolu je jeho fixná štruktúra a z toho vyplývajúca nemožnosť pridania ďalších položiek do dát tokov.

NetFlow v9

Verzia 9 zavádza vylepšenie v podobe šablón, ktoré špecifikujú obsah prenášaných dát tokov [32]. Šablóny prestavajú asi najväčší prínos oproti verzii 5 a zvyšujú tak flexibilitu tohto protokolu. Táto verzia ďalej podporuje polia pre IPv6, virtuálne lokálne siete (VLAN), MAC adresy a mnohé ďalšie. NetFlow v9 je podporovaný na väčšine Cisco smerovačoch a prepínačoch [27].

IPFIX

Protokol IPFIX je principiálne založený na protokole NetFlow v9, je však ešte viac obecnejší a flexibilnejší, preto sa niekedy označuje aj ako NetFlow v10 [27]. IPFIX je teda podobne ako NetFlow v9 založený na šablónach [35]. Motiváciou zavedenia protokolu IPFIX je zmenšenie závislosti od proprietárnych štandardov a sieťových zariadení firmy Cisco, čo je docielené obecným a flexibilným návrhom IPFIX protokolu [30].

2.3 Analýza NetFlow dát

Analýza dát predstavuje finálny a zrejme najdôležitejší krok v procese spracovania tokov. V tejto časti sú popísané tri typické prípady použitia analýzy: analýza tokov a hlásenie udalostí, detekcia hrozieb a monitorovanie výkonu.

Dáta, ktoré sú v tejto fáze analyzované, vznikli v procese zbierania tokov a sú závislé od nastavenia celej architektúry NetFlow. K procesom priamo ovplyvňujúcim dáta tokov patrí najmä vzorkovanie, filtrovanie a agregácia. Napríklad voľba deterministického vzorkovania v exportéri môže viesť ku skresleniu charakteristík periodickej komunikácie. Okrem priamo nastaviteľných parametrov ovplyvňujú výsledok zberu aj dostupné prostriedky sieťových zariadení. Do tejto kategórie môžeme radiť pamäť, výpočtový výkon a prenosovú kapacitu

medzi exportérom a kolektor. Nedostatok niektorého z týchto zdrojov môže viesť k predčasnému exportovaniu toku či jeho strate, čo má znova dopad na povahu samotných dát tokov [10].

Analýza tokov a hlásenie udalostí

K základným úkonom analýzy patrí vyhľadávanie v dátach tokov, ich filtrovanie a zobrazovanie štatistík. Typicky sa jedná o štatistiky typu *top-talkers*, teda identifikovanie tých častí sietí (zariadení, podsietí), ktoré spolu komunikujú najviac, resp. produkujú výrazne vyšší objem komunikácie ako ostatné prvky siete. Ďalším bežným prípadom použitia je zaznamenávanie udalostí (*Alerting*). K zaznamenaniu udalostí môže dôjsť napríklad po prekročení zvoleného prahu, či pri použití určitých portov (aplikácií).

Vyššie uvedenú funkcionalitu poskytuje napríklad nástroj `NfSen`[1] z rodiny nástrojov kolektoru `nfdump`. Rozhranie nástroju `NfSen` je realizované formou webovej stránky. Dôležitou časťou rozhrania, nie len nástroja `NfSen`, sú informácie sumarizujúce stav siete (*Dashboards*). Jedná sa predovšetkým o grafy prenesených tokov, paketov a bajtov v závislosti na čase. Veľké výchylky v týchto grafoch často naznačujú vznik anomálie v sieti. Situáciu je potom možné prešetriť bližšie spresnením časového okna a vhodnou voľbou filtrov. Napríklad DDoS útoky sú známe tým, že vytvarujú veľa malých, typicky jednopaketových tokov, čo sa prejaví práve zvýšením počtu tokov v čase útoku. Nástroj `NfSen` môže byť ďalej rozšírený formou zásuvných modulov napríklad za účelom automatickej detekcie hrozieb či monitorovania výkonu [10].

Detekcia hrozieb

Detekcia hrozieb v prostredí tokov má dve značné výhody oproti detekcii z paketov. Prvá z výhod predstavuje podstatne nižšia priestorová a výpočtová náročnosť spracovania tokov, čo umožňuje ich nasadenie aj vo vysokorýchlostných linkách siete. Druhá spočíva v možnosti detekcie hrozieb aj zo šifrovanej komunikácie nakoľko toky prirodzene neobsahujú informácie z tela paketu [10]. Cieľom detekcie hrozieb pomocou tokov však nie je nahradenie paketových metód. Metódy založené na tokoch dopĺňajú existujúce metódy založené na paketoch a sú použité najmä v prípadoch, kde by bola realizácia paketového prístupu ťažko realizovateľná [37].

V internetovej sieti existuje niekoľko druhov nelegálnych aktivít a útokov, ktoré môžeme deliť do nasledujúcich kategórií [37]:

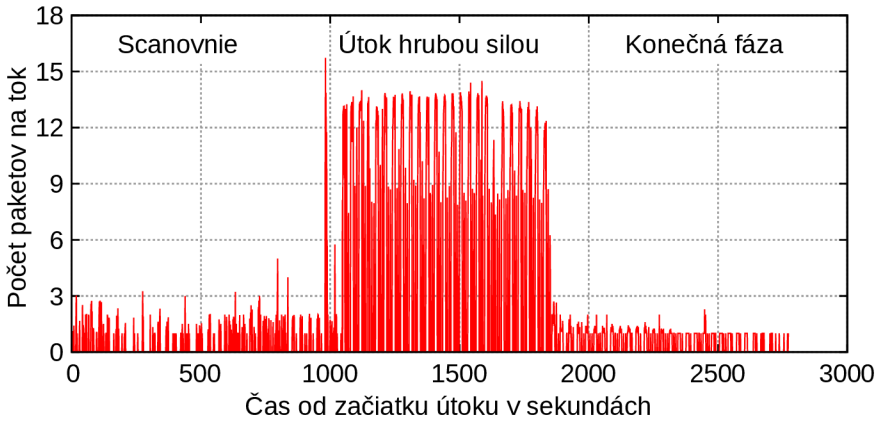
- Fyzické útoky – cieľom je fyzicky poškodiť sieťové zariadenie.
- Pretečenie pamäte – útoky sa snažia nadobudnúť kontrolu nad systémom alebo obmedziť jeho činnosť technikou pretečenia pamäte (*buffer overflow*).
- Útoky na heslo – snaha o získanie hesla neoprávneným užívateľom.
- Útoky odmietnutia služby (*DoS*) – cieľom útoku je znemožniť legitimným užívateľom prístup, resp. používanie služby.
- Scany – získavajú informácie o stave siete, ktoré sú neskôr potenciálne použité pre uskutočnenie ďalších útokov.

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

- Trójsky kôň – potenciálne škodlivý program, ktorý sa navonok tvári ako užitočný.
- Červy – škodlivý program so schopnosťou pomerne rýchlej samoreplikácie v sieti.
- Vírusy – škodlivý program, ktorý sa replikuje len v rámci infikovaného. zariadenia, replikácia na ďalšie zariadenia v sieti je závislá na akcii infikovaných užívateľov. Šíria sa pomalšie ako červy.
- Botnety – jedná sa o skupinu zariadení, ktoré boli infikované škodlivým programom. Infikované zariadenia ovláda bez vedomia vlastníkov zariadení tzv. bot master. Tieto zariadenia potom môže bot master zneužiť k realizovaniu ďalších typov útokov.

Z uvedených kategórií je možné prostredníctvom tokov detegovať najmä: scany, botnety, červy a DoS útoky. Možnosti detekcie skôr ale záležia od konkrétnej formy útoku. Napríklad pri útoku DoS rozlišujeme dve základné varianty. Prvá sa snaží zahltiť systém požiadavkami a druhá využíva chyby v návrhu či implementácii systému. Výsledkom prvej varianty je obrovské množstvo paketov a teda aj tokov, čo umožňuje detekciu tejto udalosti. V druhej variante je počet paketov výrazne nižší. Príkladom je útok *Ping of death*, ktorý spočíva vo vytvorení škodlivých paketov spôsobujúcich pád systému príjemcu. Keďže sa jedná len o niekoľko málo paketov, tento útok by s veľkou pravdepodobnosťou nebol v prostredí tokov detegovaný [37].

Samotná detekcia typicky prebieha na základe dát z niekoľkých tokov, nad ktorými sú definované rôzne metriky. Môže sa napríklad jednať o výpočet pomeru medzi počtom paketov a tokov, priemernej veľkosti paketov, resp. tokov, sledovanie počtu spojení z a do určitej IP adresy [37] či vyhľadávanie IP adries na zoznamoch podozrivých zariadení [10].



Obr. 2.3: Priebeh slovníkového útoku na program SSH zobrazený prostredníctvom metriky počtu paketov na tok od útočníka k obeti [10].

Na demonštračné účely detekcie útoku v prostredí tokov je na obrázku 2.3 zobrazený priebeh slovníkového útoku na program SSH. Obrázok zachytáva priebeh útoku voči počtu paketov v toku v závislosti na čase. Program SSH umožňuje bezpečný (šifrovaný) vzdialený prístup k zariadeniam v sieti. Tento prístup je väčšinou podmienený znalosťou hesla. Cieľom útočníka je teda nájsť správne heslo a získať tak prístup k zariadeniu. V prvej fáze útoku dochádza k vyhľadaniu potenciálnych obetí – zariadení v sieti, na ktorých beží služba SSH. Táto fáza je na obrázku označená ako scanovanie a je špecifická malým počtom paketov v tokoch. Po nájdení obete nasleduje slovníkový útok, teda skúšanie rôznych hesiel (útok

hrubou silou). V rámci jedného spojenia je uskutočnených niekoľko pokusov hádania hesla, čo spôsobí väčší počet paketov v toku. Nasleduje konečná fáza, kedy ešte môže dôjsť k výmene niekoľkých paketov medzi infiltrovanou obeťou a útočníkom. Najdôležitejším znakom tohto útoku je veľký rozdiel medzi počtom paketov v toku vo fáze scanovania a skúšania hesiel, ako je možné pozorovať aj na obrázku 2.3 v čase 1 000. Táto charakteristika útok pomerne jednoducho identifikuje aj napriek tomu, že sa jedná o šifrovanú komunikáciu [10].

Monitorovanie výkonu

Dáta tokov je taktiež možné použiť na kontrolu dohody o poskytovaní služieb (SLA) a sledovania výkonnosti služieb. Jedná sa napríklad o jednosmerné resp. obojsmerné oneskorenie služby, paketovú stratu, či prenesené dáta a mnohé ďalšie metriky. Cieľom tejto analýzy môže byť identifikovanie takých udalostí na sieti, ktoré vedú k zníženiu kvality poskytovania danej služby. V prípade metrík založených na čase počiatku a konca toku je potrebné venovať pozornosť najmä synchronizácii času na exportéroch, inak môžu byť výsledky metriky značne skreslené [10].

Kapitola 3

Charakteristika DDoS útokov

V úvode tejto kapitoly je popísané základné rozdielne útokov odopretia služby (DoS). Zvyšná časť sa venuje najmä podrobnejšiemu popisu DoS útokov využívajúcich mechanizmus podvrhnutia IP adries, ktoré sú hlavným predmetom skúmania v tejto práci. Cieľom DoS útokov je zabrániť (odoprieť) prístup k službe legitimným užívateľom [6]. Mnohé DoS útoky sú vykonané nie z jedného, ale viacerých zariadení, napríklad prostredníctvom botnetov, čím je možné zvýšiť intenzitu DoS útoku. V tom prípade sa jedná o distribuovaný útok odopretia služby (DDoS) [11, 10]. Z tohto dôvodu sa často označenie DoS zamieňa s pojmom DDoS. Preto sú v tejto kapitole ďalej označené útoky odopretia služby súhrne pojmom DDoS, aj keď nemusia byť nutne distribuované.

3.1 Rozdelenie DDoS útokov

Útoky odoprenia služby sú na základe ich povahy delené do dvoch kategórií: sémantickej a volumetrickej. Sémantický útok využíva slabosti protokolu či aplikácie. Tieto útoky sú založené na špeciálne vytvorených paketoch, ktoré môžu spôsobiť nadmerné použitie výpočtových zdrojov, spomalenie a vypnutie sieťového zariadenia. Volumetrické útoky sú oproti tomu založené na zahltení určitého prvku v sieti. Ich cieľom je teda zahltiť sieťový prvok do takej miery, aby došlo k narušeniu legitímnej komunikácie [11, 6].

Sémantickým útokom je možné predísť odhalením a opravením bezpečnostnej slabosti, ktorá útok umožňuje. Tento prístup však nie je vždy možný. Napríklad útok zahltenia SYN paketmi, popísaný nižšie, patrí principálne do kategórie sémantických útokov. Tento útok je síce možné potlačiť rôznymi mechanizmami, ak je ale intenzita tohto útoku dostatočne veľká môže byť aj napriek tomu úspešný. Preto sa charakter tohto útoku mení zo sémantického na volumetrický. Tento príklad zároveň ilustruje väčšiu problematickosť ochrany pred volumetrickými útokmi, nakoľko sú jeho obeťami aj systémy, ktoré sú z pohľadu návrhu dobre zabezpečené [6, 25]. Volumetrické útoky sú pre preto pomerne bežné a ich počet stále rastie. V týchto útokoch navyše často dochádza k náhodnému podvrhnutiu IP adries útočníkov, resp. útočníka, aby došlo k zamedzeniu jeho odhalenia [11].

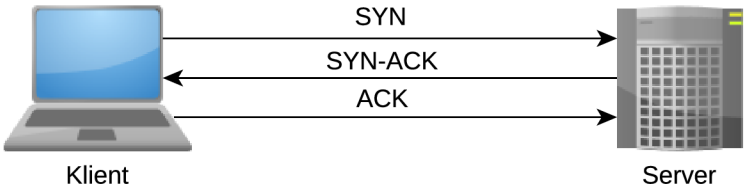
Typickým príkladom sémantického útoku, aj keď dnes už zastaralého, je *Ping of Death*. Tento útok je založený na zaslaní špecificky zostavených paketov, ktoré na príjemcovi spôsobia pretečenie pamäte čím môže k dôjsť pozastaveniu alebo pádu daného systému [21].

Zvyšná časť tejto kapitoly sa sústreďuje na volumetrické útoky nakoľko sú najbežnejším prípadom DDoS útokov na internete [29]. Nižie sú osobitne uvedené typické volumetrické útoky: SYN, UDP a ICMP zahltenie. Mnohé z týchto útokov využívajú určitú vlasnosť

protokolu, čo ich radí skôr k sémantickým útokom, avšak vzhľadom na typicky vysokú intenzitu týchto útokov sú v tejto práci klasifikované ako volumetrické.

TCP (SYN) zahltenie

Tento útok v súčasnosti predstavuje najbežnejšiu formu DDoS útoku na internete [29, 11]. SYN zahltenie je založené na zneužití mechanizmu ustanovenia spojenia v transportnom protokole TCP, kedy dochádza k výmene troch paketov (*three way handshake*). V bežnej situácii klient nadviaže spojenie zaslaním TCP paketu s príznakom SYN, server následne odpovie paketom s príznakmi SYN-ACK a zároveň alokuje prostriedky pre dané spojenie. Klient následne potvrdí ustanovenie spojenia zaslaním paketu s príznakom ACK. Tento proces je zobrazený na obrázku 3.1. Samotný útok spočíva v zaslaní veľkého množstva SYN paketov. Server tieto pakety interpretuje ako pokus o nadviazanie spojenia, vyhradí pre ne prostriedky a zašle späť paket s príznakmi SYN-ACK. Útočník však ustanovenie spojení nedokončí, čím ostávajú v tzv. polo-otvorenom stave (*half-open*) a prostriedky, ktoré server týmto spojeniam vyhradil sú uvoľnené až po uplynutí určitého časovača. Ak je tento útok dostatočne intenzívny dôjde k zahlteniu prostriedkov serveru a teda odmietnutiu legitímnych klientov [25].



Obr. 3.1: Proces ustanovenia spojenia medzi klientom a serverom (*three-way handshake*).

V tomto type útoku, ako aj v prípade UDP či ICMP zahltenia, navyše útočník často maskuje svoju identitu podvrhnutím zdrojovej IP adresy [11, 1]. Ak by totiž útočník zdrojovú adresu nezmenil riskuje svoje odhalenie a prípadné potlačenie útoku. V prípade použitia niekoľkých zariadení na prevedenie útoku prostredníctvom botnetu nie je nutné, aby útočník použil podvrhnutie IP adries nakoľko útočné pakety zasielajú infikované zariadenia a nie samotný útočník. Ak chce ale útočník ešte viac zneprehľadniť povahu útoku, môže tieto prístupy kombinovať. To znamená, že infikované zariadenia botnetu budú navyše podvrhovať IP adresy v útočných paketoch [25].

Jednoduchou technikou ako sa brániť pred týmto typom útoku je zväčšenie dostupných prostriedkov na serveri alebo postupne nahrádzať (recyklovať) najstaršie polo-otvorené spojenia novými. Tento prístup však len zvyšuje hranicu odolnosti služby, a ak má útok dostatočnú intenzitu, dôjde znova k odopretiu prístupu legitímnych užívateľov. Pokročilejšou technikou obrany je použitie tzv. *SYN cookies*, kedy server nealokuje pre polo-otvorené spojenia prostriedky, ale potrebné informácie pre úspešné ustanovenie spojenia uloží do paketu SYN-ACK, ktorý zasiela spätne klientovi. Klient následne zašle paket ACK so všetkými potrebnými informáciami pre ustanovenie spojenie. V prípade útočníka, a to najmä v vtedy, ak je použitá podvrhnutá zdrojová IP adresa, nedôjde k zaslaniu finalizujúceho ACK paketu a teda nie sú alokované ani prostriedky na serveri. *SYN cookies* sú navrhnuté tak, aby boli kompatibilné so štandardnou implementáciou protokolu TCP. Kompatibilita ale spôsobuje limitovanie objemu informácií, ktoré je možné do SYN-ACK odpovede za-

kódovať. To znamená, že môže dôjsť k strate určitej informácie o spojení, čo predstavuje nevýhodu tohto prístupu [25]. Mechanizmus *SYN cookies* je taktiež z výpočtového hľadiska pomerne náročný a pridáva dodatočnú zátaž na server. Ďalšou nevýhodu tohto prístupu je problematické nadviazanie spojenia s legitimným užívateľom v prípade, ak dôjde k strane ACK paketu, nakoľko práve tento paket obsahuje potrebné informácie o spojení. Server v tejto situácii nevie detegovať stratu tohto paketu pretože si o spojení neuchováva žiadne informácie. Klient po odoslaní tohto paketu zároveň predpokladá úspešné nadviazanie spojenia a čaká dáta od serveru. V štandardnom prípade bez použitia *SYN cookies* by pri strate ACK paketu došlo po určitom čase k retransmisii SYN-ACK paketu, na čo by klient znova zaslal ACK paket a komunikácia by mohla pokračovať [28]. Ďalším možným riešením obrany pred týmto útokom je použitie medzi-serveru (*proxy*), ktorý oddeľuje útočníka od služby. Princíp proxy serveru spočíva v tom, že k službe prepustí len úspešne ustanovené spojenia, čím prenáša réžiu spojenú s obranou proti DDoS útokom na seba [25].

UDP zahltenie

V tomto type útoku sú obeti zasielané pakety prostredníctvom transportného protokolu UDP. Protokol UDP je narozdiel od protokolu TCP bezstavový, to znamená, že si neudržuje žiadnu informáciu o spojení. Pri obdržaní UDP paketu musí preto príjemca vyhľadať aplikáciu identifikovanú tzv. číslom portu. Toto vyhľadanie so sebou prináša určitú výpočtovú réžiu. Útočník túto vlasnosť využíva tak, že obeti zasiela UDP pakety s náhodnými číslami portov. Pre každý z týchto paketov sa následne vyhľadáva odpovedajúca aplikácia a dochádza tak ku nadmernému konzumovaniu výpočtových prostriedkov, ak navyše nie je port asociovaný so žiadnou aplikáciou dôjde k vytvoreniu ICMP paketu, ktorý odosielateľa informuje o nedostupnosti cieľa (aplikácie). Aby útočníci vyťažili prostriedky servera čo najviac, môžu byť tieto pakety pomerne veľké. Podobne ako v prípade SYN zahltenia útočníci utajujú svoju identitu podvrhnutím IP adries alebo použitím botnetu [26].

Špeciálnym prípadom podvrhnutia IP adries je tzv. amplifikačný útok. V tomto prípade nahradí útočník svoju IP adresu (zdrojovú) adresou obete útoku. Takto podvrhnuté pakety zasiela útočník na tzv. reflektory, teda zariadenia, na ktorých bežia aplikácie s amplifikačným charakterom – ich odpoveď je oveľa väčšia ako požiadavka. Tieto odpovede sú však zaslané obeti nakoľko bola IP adresa podrhnutá [26]. Útočník vie týmto spôsobom výrazne znásobiť objem útoku. Typickými aplikačnými protokolmi, ktoré amplifikáciu umožňujú a ako transportný protokol používajú UDP sú DNS a NTP. V prípade DNS môže byť amplifikačný faktor, teda pomer veľkosti žiadosti a odpovede 1:70 a v prípade NTP až 1:200 [23]. Amplifikácia je ale skôr obecným mechanizmom útoku a neviaže sa ku konkrétnemu protokolu. Protokol UDP je však vďaka svojej bezstavovosti vhodným kandidátom na použitie tohto mechanizmu, nakoľko nevyžaduje overenie IP adresy (ustanovenie spojenia), a ak daná aplikácia nevykonáva dodatočné overenie zdroja je pomerne ľahké presmerovať odpovede na obeť. V prípade protokolu TCP by dôsledkom zmeny zdrojovej IP adresy nedošlo k ustanoveniu spojeniu a podvrhnutá IP adresa by obdržala len odpoveď serveru na pokus o ustanovenie spojenia vo forme paketu s príznakmi SYN-ACK.

Obrana proti UDP zahlteniu je pomerne náročná a vyžaduje inšpekciu obsahu paketu (*deep packet inspection*) [26].

ICMP zahltenie

Protokol ICMP slúži hlavne na hlásanie chýb vzniknutých na sieti a overenie pripojenosti zariadení. Pri ICMP útoku je použitá tzv. správa typu žiadosť o odpoveď (*Echo request*),

ktorá za normálnych podmienok slúži na overenie konektivity zariadenia. Po obdržaní tejto správy zasiela príjemca odpoveď (*Echo reply*), čím je napríklad možné zistiť časové oneskorenie medzi dvoma zariadeniami (*round trip time*). V prípade útoku útočník na odpoveď nečaká a zasiela žiadosti opakovane, aby došlo k preťaženiu prostriedkov obete. Podobne ako v predošlých prípadoch môže útočník svoju IP adresu podvrhnúť alebo na uskutočnenie použiť niekoľko zariadení formou botnetu. Prevencia tohto útoku je možná obmedzím intenzity príjmu ICMP paketov alebo limitovaním použitia ICMP žiadostí len na určité podsiete [24].

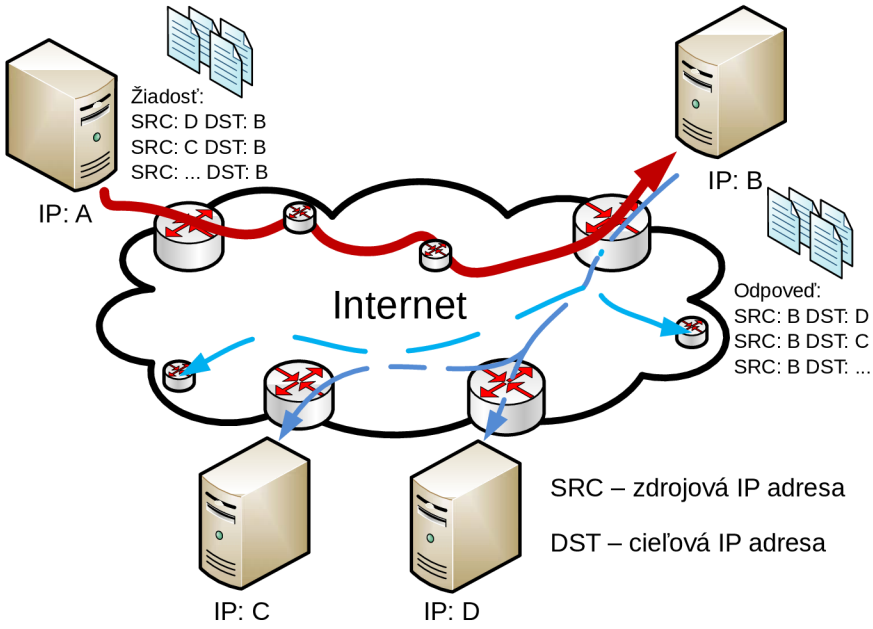
3.2 Podvrhnutie IP adresy v útočnom pakete

Podvrhnutie IP adries v DDoS útokoch je pomerne bežnou technikou utajenie útočníka [11, 25]. Adresy sú väčšinou podvrhnuté náhodne [17]. Výnimkou je amplifikačný útok, v ktorom je IP adresa podvrhnutá tak, aby útočné pakety dorazili od reflektoru k obeti [11]. Nenáhodné podvrhnutie zdrojovej IP adresy môže byť tiež použité na presmerovanie viny z útoku od útočníka na legitimného užívateľa. V tom prípade je legitimný užívateľ neprávom označený službou, na ktorú bol útok cielený ako útočník, čo môže ďalej viesť k zablokovaniu prístupu legitimného užívateľa k tejto službe [1, 31].

Náhodnosť podvrhnutia IP adries teoreticky umožňuje pozorovať útok v rámci celého internetu. Toto pozorovanie je možné uskutočniť prostredníctvom tzv. spätne rozptýlených paketov (*backscatter*). Sú to pakety, ktoré dorazia na podvrhnuté IP adresy od obete útoku. Spätne rozptýlené pakety sú ilustrované na obrázku 3.2. Na tomto obrázku sú útočníkove pakety znázornené červene a mieria na obeť s IP adresov B. Obeť na tieto pakety reaguje zaslaním odpovede (modrá). Pakety odpovede však nesmerujú späť k útočníkovi s IP adresou A, ale k zariadeniam s IP adresou C a D, ktoré boli podvrhnuté útočníkom. Na zariadeniach C a D je teda možné pozorovať pakety od obete útoku.

Princíp rozptýlených paketov a náhodnosť podvrhnutia využívajú tzv. teleskopy. Teleskopy sú tvorené pomerne veľkým nepoužitým adresovým blokom IP adries. Na teleskopoch nie sú typicky žiadne legitímne zariadenia. Príchod paketu na adresový rozsah teleskopu preto indikuje priebeh nelegitímnej aktivity. Typicky sa jedná o DDoS útok či scan, niekedy ale pakety dorazia na teleskop chybou konfigurácie siete. Vďaka tejto vlastnosti je detekcia DDoS útokov v prostredí teleskopu pomerne priamočiara a typicky založená na rôznych kvantitatívnych prahoch, ktoré sú schopné odlíšiť DDoS útoky od scanov a komunikácie vzniknutej chybami v konfigurácii sieťových prvkov. Vhodným príkladom detekcie DDoS útoku na teleskope je útok zahltenia SYN paketmi. Spätne rozptýlené pakety tohto útoku je na teleskope možné jednoducho identifikovať prostredníctvom príznakov paketu SYN-ACK oproti tomu pakety pochádzajúce zo scanov budú mať len príznak SYN. Potom už len stačí zvoliť vhodný prah na počet a intenzitu obdržaných rozptýlených paketov, aby bola väčšia istota toho, že sa skutočne jedná o DDoS útok [17, 1]. Podrobnejšie sú metódy detekcie DDoS útokov zo spätne rozptýlených paketov rozobrané v kapitole 5. Prirodzene čím väčší je teleskop tým rýchlejšie a spoľahlivejšie je DDoS útok detegovaný. Vyhodnotenie vzťahu veľkosti teleskopu, resp. monitorovacej siete voči možnostiam detekcie DDoS útokov z rozptýlených paketov je diskutované v kapitole 4.

Na detekciu amplifikačných DDoS útokov je možné použiť tzv. honeypoty. Honeypot je sieťové zariadenie, ktorého cieľom je prilákať útočníkov a tým získať znalosti o útoku a správaní útočníkov. Amplifikačné honeypoty sú navrhnuté tak, aby sa navonok tvárili ako vhodné reflektory a útočníci ich použili k uskutočneniu útoku. Narozdiel od skutočných reflektorov je však miera odosielania odpovedí z honeypotu limitovaná, aby príliš neprispeli



Obr. 3.2: Ukážka spätne rozptýlených paketov útoku (modrá). Útočník (A) vykonáva útok (červená) na zariadenie B. Útočník svoju zdrojovú adresu podvrhol za IP adresy C a D. Obeť útoku B preto zasiela odpoveď (modrá) na požiadavok útočníka A na IP adresy C a D [1].

k samotnému útoku. Útočník použitým týchto honeypotov odhalí svoju stratégiu a aj to na koho je DDoS útok smerovaný [12]. Narozdiel od teleskopov, ktoré sú navrhnuté pre sledovanie DDoS útokov s náhodným podvrhnutím IP adries a potrebujú pomerne veľký adresový priestor [16], stačí na sledovania amplifikačných DDoS útokov len pár honeypotov, nakoľko ich útočníci aktívne vyhľadávajú pomocou scanovania siete [12].

Hoci sú dnes DDoS útoky s náhodne podvrhnutými adresami pomerne bežné [11], táto metóda nie je použitá vždy. Útočník môže totiž realizovať útok pomocou botnetu a v tom prípade je už skrytý za skupinou infikovaných zariadení, ktoré útok vykonávajú. Motivácia dodatočného podvrhnutia IP adries pri použití botnetu potom nie je tak veľká, ako keď útok pochádza priamo z útočníkovho zariadenia [1, 25]. Nevýhodou tejto metódy podvrhnutia a zároveň prevenciou pred DDoS útokmi s náhodne podvrhnutými paketmi je zavedenie filtrovania paketov na úrovni podsietí. Toto filtrovanie funguje tak, že odchod paketov z podsiete je povolený len pre tie pakety, ktoré z nej pochádzajú. Útočníkove možnosti podvrhnutia IP adries sú týmto obmedzením značne limitované a podvrhnutie je možné len v rámci podsiete, v ktorej sa útočník sám nachádza čo zároveň uľahčuje jeho prípadné vypátranie [31]. Hoci je tento spôsob filtrovania doporučený (*best current practise*) už od roku 2000, DDoS útoky s podvrhnutými IP adresami sú stále možné a bežné. Organizácia CAIDA preto priebežne monitoruje možnosti podvrhnutia IP adries na internete prostredníctvom projektu Spoofer[1]. Monitorovanie prebieha najmä vďaka dobrovoľníkom z rôznych častí sveta, ktorí majú spustený program overujúci možnosti podvrhnutia v ich podsieti. CAIDA následne periodicky zhromažďuje takto získané dáta od dobrovoľníkov. Podľa výsledkov z tohto projektu bolo v roku 2019 možné vykonať podvrhnutie zdrojovej IP adresy z viac ako 1/4 autonómnych systémov, ktoré boli do výskumu zapojené. CAIDA taktiež vyhodnocovala

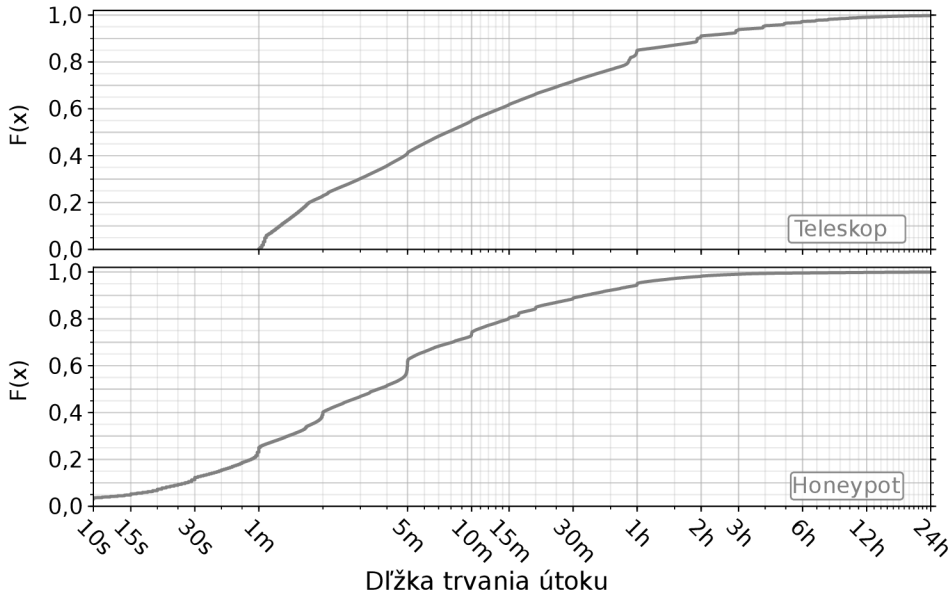
[1] https://www.caida.org/projects/spoofer/

náchylnosť k vstupu podvrhnutého paketu do autonómneho systému, z ktorého paket nepochádza. V tomto prípade boli výsledky ešte horšie a k prepusteniu takýchto paketov došlo v 2/3 testovaných autonómnych systémov [13].

3.3 Kvantifikovanie DDoS útokov na internete

Organizácia CAIDA sa pomerne intenzívne venuje výskumu a kvantifikácii DDoS útokov. V roku 2017 publikovali článok [11], v ktorom analyzujú takmer dva roky (731 dní) dát zozbieraných od 3.1.2015 do 2.28.2017. Dáta pochádzajú z dvoch zdrojov, prvým je pomerne veľký CAIDA teleskop s prefixom siete /8 a druhým zdrojom je amplifikačná sieť honeypotov AmpPot [12], ktorú v čase zbierania dát tvorilo 24 amplifikačných honeypotov. Teleskopy zachytávajú DDoS útoky s náhodne podvrhnutými IP adresami a honeypoty amplifikačné DDoS útoky.

V priemere je každý deň uskutočnených takmer 30 000 DDoS útokov, z toho 17 100 využíva náhodné podvrhnutie a 11 600 má amplifikačný charakter. Útočníci navyše často zároveň kombinovali oba typy útokov. Počas týchto dvoch rokov bola aspoň raz napadnutá až tretina všetkých aktívnych podsietí s prefixom /24. Z hľadiska typu útokov bolo na teleskope pozorované najmä TCP zahltenie a to v 78,4 % útokov, za ktorým s pomerne veľkým rozdielom nasleduje UDP zahltenie (15,9 %), ICMP zahltenie (4,5 %) a ostatné útoky (0,2 %). Z pohľadu služieb boli cieľom najmä webové stránky, online hry a MySQL servery. Amplifikačné DDoS útoky používali na ich realizáciu protokoly, respektíve služby: NTP (40,08 %), DNS (26,17 %), CharGen (22,37 %), SSDP (8,38 %), RIPv1 (2,27 %) a iné (0,73 %) [11].

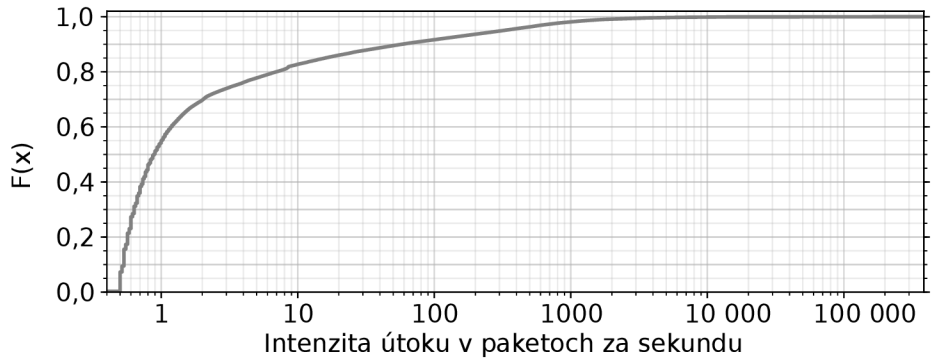


Obr. 3.3: Graf zobrazujúci distribučnú funkciu dĺžky trvania útokov v rámci pozorovania na teleskope (hore) a honeypotoch (dole) [11].

Na obrázku 3.3 je zobrazená distribučná funkcia dĺžky trvania útokov pre útoky pozorované na teleskope a honeypotoch. V prípade teleskopu je najnižšia hodnota trvania jedna minúta nakoľko zvolená metóda detekciu kratších útokov nedovoľuje. Z grafov je možné pozorovať, že väčšina DDoS útokov netrvá dlhšie ako 8 minút. Dlhé útoky sú pomerne vzácne.

V prípade náhodne podvrhnutých útokov trvá viac ako 1,5 hodiny len 10 % útokov. Podobne v prípade amplifikačných útokov tvoria horných 10 % distribúcie útoky, ktoré trvajú 40 a viac minút. Priemerná dĺžka trvania útokov, resp. medián na teleskope predstavuje 48 minút, resp. 454 sekúnd a na honeypotoch 18 minút, resp. 255 sekúnd.

Na obrázku 3.4 je ďalej zobrazená intenzita DDoS útokov na teleskope. Pre získanie skutočnej hodnoty intenzity je potrebné hodnoty v grafe vynásobiť číslom 256 nakoľko bola intenzita meraná na teleskope s prefixom siete /8. Z uvedeného grafu je možné pozorovať, že väčšina útokov má pomerne malú intenzitu. Konkrétne až 70 % útokov na teleskope má intenzitu menšiu ako 512 paketov za sekundu, čo pri veľkosti paketu 40B odpovedá 163,84 kb/s a len 17 % útokov má intenzitu väčšiu ako 2 560 paketov za sekundu (820 kb/s pre 40B paket). V prípade amplifikačných útokov je presný odhad intenzity útokov narozdiel od teleskopu problematickejší, nakoľko je reflektorov typicky niekoľko a honeypot môže byť len jedným z nich. Autori [11] však uvádzajú, že amplifikačné útoky sú intenzívnejšie ako náhodne podvrhnuté. Odhad intenzity útoku na teleskope je jednoduchší najmä vďaka predpokladu náhodnosti podvrhnutia, intenzita je potom daná počtom pozorovaných rozptýlených paketov a veľkosťou teleskopu [16].



Obr. 3.4: Graf zobrazujúci distribučnú funkciu intenzity DDoS útokov viditeľných na teleskope CAIDA v paketoch za sekundu. Uvedené hodnoty predstavujú intenzitu paketov smerujúcich na teleskop, skutočnú intenzita útoku je možné získať vynásobením hodnotou 256 [11].

Kapitola 4

Štatistický model spätného rozptylu

V tejto kapitole sú popísane najmä teoretické možnosti detekcie DoS útokov na základe spätného rozptylu. Z uvedených štatistických modelov je asi najdôležitejším záverom určenie vzťahu medzi veľkosťou teleskopu, resp. podsiete a počtom pozorovaných rozptýlených paketov útoku. Tento vzťah predurčuje aké množstvo útokov a ako presne je útoky možné pozorovať. Pre porovnanie sú v príkladoch tejto kapitoly použité štyri rôzne veľkosti podsietí: teleskop organizácie CAIDA /8 [5], odporúčaná veľkosť teleskopu na základe článku z TU Delft /17 [1], skutočne použitý rozsah v meraniach v článku od TU Delft /15 [1] a rozsah organizácie CESNET / 12, pričom za daným názvom je uvedený počet bitov prefixu siete. Pri výpočtoch a príkladoch je uvažovaný adresový priestor IPv4, ktorý obsahuje 2^{32} rôznych IP adries. Celkový počet IP adries podsiete je teda daný ako $2^{32-prefix}$. V prípade organizácie CESNET, ktorá sa skladá z niekoľkých podsietí, bol prefix zaokrúhlený na najbližšie celé číslo. Absolútne hodnoty v počte IP adries, ktoré spadajú pod danú podsieť, sú zobrazené v tabuľke 4.1. V prípade organizácie CESNET bol v tejto kapitole pri výpočtoch vždy použitý skutočný počet adries nie zaokrúhlený.

Väčšina štatistických odvodení prezentovaných v tejto kapitole bolo prevzatých z článkov [16, 17] od autorov organizácie CAIDA.

Predpoklad štatistického odvodenia

Odvodenie štatistických vlastností DDoS útokov na základe spätne rozptýlených paketov je založené hlavne na predpoklade náhodnosti podvrhnutia zdrojovej IP adresy v útočnom pakete. Nasledujúce analýzy uvažujú pri štatistickom odvodení rovnakú pravdepodobnosť

Podsieť	Počet IP adries
CAIDA /8	16 777 216
CESNET /∼12	1 048 576* (913 408)
TU Delft /15	131 072
TU Delft /17	32 768

Tabuľka 4.1: Porovnanie počtu IP adries v rôznych podsietiach v závislosti od veľkosti prefixu siete. *Uvedené číslo je zaokrúhlené na najbližší celočíselný prefix siete v zátvorke je uvedený skutočný počet IP adries.

výberu podvrhnutej IP adresy a ich vzájomnú nezávislosť. Výber IP adresy teda predstavuje Bernoulliho pokus, počet pozorovaných paketov na podsieti sa potom riadi Binomickým rozložením pravdepodobnosti. Predpoklad náhodnosti vychádza z implementácie existujúcich nástrojov na tvorbu DDoS útokov a bol taktiež empiricky overený pomocou štatistického testu Anderson–Darling [17].

4.1 Odhad veľkosti útoku

Pravdepodobnosť pozorovania náhodne podvrhnutého paketu je určená pomerom veľkosti monitorovanej podsiete k celému priestoru IP adries. Formálne pravdepodobnosť p pozorovania paketu s náhodne podvrhnutou IP adresou, respektíve pravdepodobnosť p úspechu v notácii Bernoulliho pokusu s veľkosťou podsiete k IP adries je:

$$p = \frac{k}{2^{32}} \tag{4.1}$$

Z binomického rozdelenia je potom pomerne priamočiaro možné určiť očakávaný počet pozorovaní podvrhnutých paketov na podsieti o veľkosti k, ak bol celkový počet útočných paketov m:

$$E(X) = mp = \frac{km}{2^{32}} \tag{4.2}$$

Celkový počet útočných paketov, je však premenná, ktorú chceme vypočítať a naopak známy je počet pozorovaní $E(X)' \approx E(X)$ na podsieti a teda pre veľkosť útoku platí:

$$m = \frac{E(X)'}{p} = E(X)'\frac{2^{32}}{k} \tag{4.3}$$

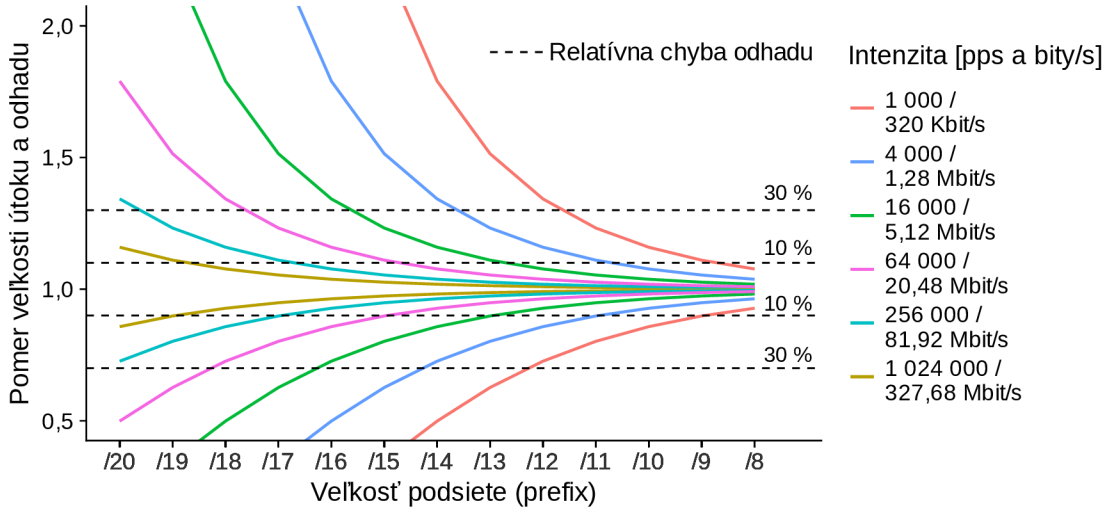
Ako je možné intuitívne tušiť, čím väčší je útok resp. podsieť, tým menší bude rozdiel medzi očakávaným počtom pozorovaných paketov $E(X)$ a skutočne pozorovaným počtom paketov $E(X)'$ na podsieti. V článku [1] uvádzajú, že siete s prefixom /17 sú dostačujúce a vedú k zhruba 2 % chybe odhadu veľkosti útoku pri intenzite 20 Mb/s. Ako najmenší použiteľný prefix autori stanovili podsieť /19, ktorá dosahuje chybovosť pod 10 % aj pri menších útokoch. Siete s prefixom /20 a viac teda nie je vhodné použiť na sledovanie spätne rozptýlených paketov. Autori stanovili tieto hranice na základe ich pozorovaní na podsieti /15 v kombinácii s výpočtom štatistickej chyby odhadu pomocou binomického rozdelenia.

Interval spoľahlivosti binomického rozdelenia je možné určiť nasledovne [15]:

$$\begin{aligned} p_h &= 1 - BetaInv(\frac{1-\alpha}{2}, n-k, k+1) \\ p_d &= 1 - BetaInv(1-\frac{1-\alpha}{2}, n-k\ +1, k+1) \end{aligned} \tag{4.4}$$

Kde n predstavuje celkový počet paketov útoku, k nameraný počet paketov spätného rozptylu, α je interval spoľahlivosti a $BetaInv$ je inverzná distribučná funkcia Beta rozdelenia. Výsledkom je horný p_h a spodný p_d odhad pravdepodobnosti p z výrazu 4.1. Tento vzťah nepriamo ohraničuje rozsah veľkosti podsietí, na ktorých by bolo pozorovaných práve k paketov s intervalom spoľahlivosti α. Teda pri spätnom prevode cez 4.3 dostávame z p_h a p_d odhad chyby v počte paketov pre danú podsieť o veľkosti k. Ako príklad uvažujme dve podsiete s prefixom /17 a /12 a útok s dĺžkou trvania 3 minúty s intenzitou 1 000 pps (paketov za sekundu). Prvá podsieť by s uvažovaním 95 % intervalu spoľahlivosti odhadla

útok v rozmedzí 59 až 4 519 pps a druhá /12 od 726 do 1 343 pps. V prípade druhej siete bude teda odhad výrazne presnejší. Na obrázku 4.1 sú pre tento príklad taktiež znázornené ďalšie rôzne veľkosti podsiete a útokov. Pričom interval, v ktorom sa odhad môže pohybovať je vyjadrený pomerom hornej/dolnej hranice odhadu k skutočnej intenzite. Príklad tohto pomeru z predošlej úlohy pre sieť /12 môžeme vyjadriť ako: 726/1 000 (spodný) a 1 343/1 000 (horný). Na príklade z obrázku 4.1 sa teleskopy dopúšťajú väčšej chyby ako v článku [1]. Táto nezhoda je zrejme spôsobená zvolením inej dĺžky trvania útoku, prirodzene čím je útok dlhší tým je presnosť odhadu na teleskopoch väčšia. V uvedenom príklade bol uvažovaný pomerne krátky útok s trvaním len 3 minúty. Pri uvažovaní hodinového útoku s intenzitou 1 000 pps stanoví podsieť s prefixom /17 intenzitu tohto útoku niekde medzi 661 až 1 450 pps, čo je vo výraznom kontraste s odhadom 59 až 4 519 pps pri trojminútovom útoku.



Obr. 4.1: Zobrazenie pomeru horného/dolného odhadu intenzity voči skutočnej intenzite útoku s uvažovaným intervalom spoľahlivosti 95 %. Trvanie každého útoku je 3 minutý. Pri odvodení veľkosti útoku v bitoch bola uvažovaná minimálna veľkosť TCP paketu (40B). Horizontálne čiary ohraničujú relatívnu chybu odhadu.

4.2 Pravdepodobnosť detekcie jedného paketu

Pravdepodobnosť detekcie jedného paketu útoku sa zdá byť nevýznamná. Táto pravdepodobnosť však pomerne dobre približuje dramatické rozdiely v závislosti na veľkosti použitej podsiete. Keďže útoky typicky trvajú určitý čas, bude veľkosť útoku označená ako súčin rT, kde r predstavuje počet paketov útoku odoslaných za sekundu (*paket rate*) a T dĺžku trvania útoku. Pravdepodobnosť pozorovania aspoň jedného paketu je potom daná ako:

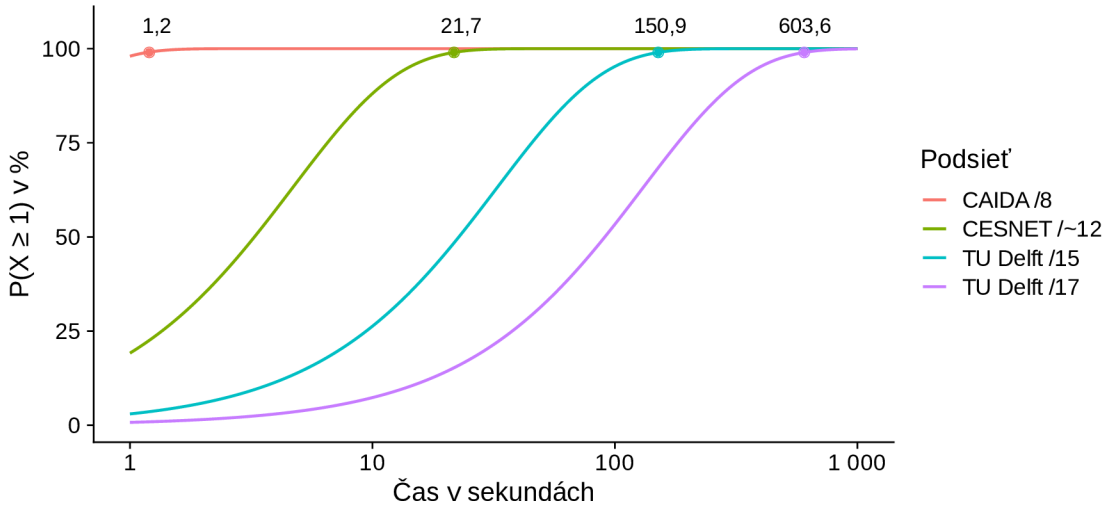
$$P(X \geq 1) = 1 - P(X = 0) = 1 - \binom{rT}{0} \cdot p^0(1-p)^{rT-0} = 1 - (1-p)^{rT} \qquad (4.5)$$

Uvažujme útok s intenzitou 1 000 paketov za sekundu, ktorý trvá 3 minúty. Očakávaný počet pozorovaných paketov tohto útoku pre rôzne podsiete zobrazuje tabuľka 4.2. Uvedené čísla však nehovoria nič o tom kedy útok začal. Obrázok 4.2 preto znázorňuje závislosť času a pravdepodobnosti videnia aspoň jedného paketu na danej podsieti. Zvýraznené body ukazujú, kedy s 99 % pravdepodobnosťou uvidíme na danej podsieti aspoň jeden paket

Podsieť	Očakávaný počet pozorovaných paketov
CAIDA /8	703
CESNET /∼12	38
TU Delft /15	5
TU Delft /17	1

Tabuľka 4.2: Očakávaný počet pozorovaných spätne rozptýlených paketov pri útoku s intenzitou 1 000 paketov za sekundu a dĺžkou trvania 3 minúty na jednotlivých podsieťach.

útoku. Zatiaľ čo pri teleskope CAIDA uvidíme útok takmer okamžite, len po zhruba jednej sekunde, v prípade najmenšej zvolenej podsiete TU Delft /17 je to 10 minút, čo je výrazne viac ako samotná dĺžka trvania uvedeného útoku. Podsieť TU Delft /17 je teda na detekciu takéhoto útoku nedostačujúca.



Obr. 4.2: Graf zobrazujúci pravdepodobnosť pozorovania aspoň jedného paketu v danej podsieti v závislosti na uplynulom čase od začiatku útoku. Intenzita útoku je 1 000 paketov za sekundu. Na každej z individuálnych kriviek podsiete je zobrazený časový bod, v ktorom dosahuje pravdepodobnosť pozorovania aspoň jedného paketu 99 %.

Na prvý pohľad sa zdá, že rozdiely medzi veľkosťou podsiete a časom pozorovania majú lineárnu závislosť. Daný pomer sa však riadi podľa výrazu:

$$\begin{aligned} 1-(1-p_1)^{rT_1} &= P = 1-(1-p_2)^{rT_2} \\ T_1 &= T_2 \frac{ln(1-p_2)}{ln(1-p_1)} \end{aligned} \tag{4.6}$$

Ak na základe vzorca 4.6 porovnávame podsieť /8 s podsieťou /24 zistíme, že v druhom prípade bude trvať detekcia aspoň jedeného paketu útoku s rovnakou pravdepodobnosťou ako na podsieti /8 o 65 664-krát dlhšie, pričom podsieť /8 obsahuje oproti /24 o 65 536-krát viac IP adries. Výsledok je teda mierne väčší ako proporcia počtu IP adries týchto podsietí.

Zo vzorca 4.5 je ďalej možné vyjadriť čas vzhľadom na určitú požadovanú pravdepodobnosť Z. Takto získaný čas bude zároveň predstavovať percentil vzhľadom na zvolenú pravdepodobnosť Z. Teda ak zvolíme hodnotu Z na 95% potom hodnota T predstavuje

Označenie podsiete	Prefix podsiete	95-ty percentil	Priemer	Medián	5-ty percentil
CAIDA	8	765,4 ms	256,0 ms	177,1 ms	13,1 ms
	12	12,3 sek.	4,1 sek.	2,8 sek.	210,1 ms
CESNET	∼12	14,1 sek.	4,7 sek.	3,3 sek.	241,3 ms
TU Delft	15	1,6 min.	32,8 sek.	22,7 sek.	1,7 sek.
	16	3,3 min.	1,1 min.	45,4 sek.	3,4 sek.
TU Delft	17	6,5 min.	2,2 min.	1,5 min.	6,7 sek.
	20	52,4 min.	17,5 min.	12,1 min.	53,8 sek.
	22	3,5 hodiny	1,2 hodín	48,5 min.	3,6 min.
1 IP adresa	32	5 mesiacov	1,7 mesiaca	1,1 mesiaca	2,5 dní

Tabuľka 4.3: Vyhodnotenie podsietí vzhľadom na čas detekcie aspoň jedného podvrhnutého paketu útoku s intenzitou 1 000 paketov za sekundu. Výpočet neuvažuje oneskorenie spôsobené prenosom po sieti.

95-ty percentil. Formuly 4.7 uvádzajú výpočet T pre danú pravdepodobnosť Z a taktiež výpočet priemernej hodnoty T, μ_T, vzhľadom na pravdepodobnosť úspešného pozorovania p. V tabuľke 4.3 je zobrazené porovnanie rôznych podsietí a ich schopnosť detegovať aspoň jeden paket útoku vzhľadom na čas T pre útok s intenzitou 1 000 paketov za sekundu. Ako je z tabuľky vidieť na teleskop CAIDA dorazí prvý paket útoku vo väčšine prípadov za menej ako sekundu bez uvažovania prípadného oneskorenia prenosu. V prípade CESNETu je to 14,1 sekundy a pri TU Delft 1,6 resp. 6,5 minút.

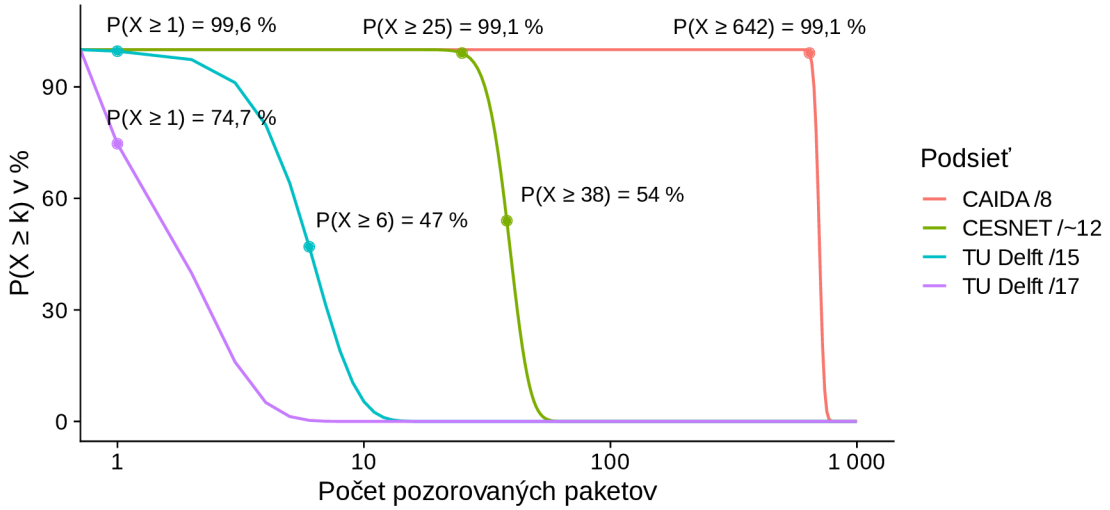
$$\begin{aligned} T &= \frac{log(1-Z)}{r * log(1-p)} \\ \mu_T &= \frac{1}{rp} \end{aligned} \tag{4.7}$$

4.3 Pravdepodobnosť detekcie niekoľkých paketov

Pre praktické účely je nutné detegovať niekoľko paketov, aby bolo útok možné klasifikovať a oddeliť od zvyšnej komunikácie. Pravdepodobnosti pozorovania aspoň k paketov útoku z celkového počtu N je:

$$P(X \geq k) = 1 - \sum_{y=0}^{k-1} \binom{N}{y} p^y (1-p)^{N-y} \tag{4.8}$$

Na obrázku 4.3 je zobrazený priebeh tejto pravdepodobnosti pre útok s intenzitou 1 000 paketov za sekundu a trvaním tri minúty. Na obrázku je možné pozorovať, že CESNET obdrží s 99,1 % pravdepodobnosťou 25 spätne rozptýlených paketov a CAIDA 642. V prípade TU Delft /15 je pozorovaný s viac ako 99,6 % šancou aspoň jeden paket a podsieť TU Delft /17 by s 25,3 % pravdepodobnosťou nepozorovala ani jeden paket útoku. Veľkosť siete teda neovplyvňuje len schopnosť detekcie útoku, ale aj počet paketov, ktoré je možné pozorovať, čo má dopad na presnosť odvodenia charakteristík útoku.

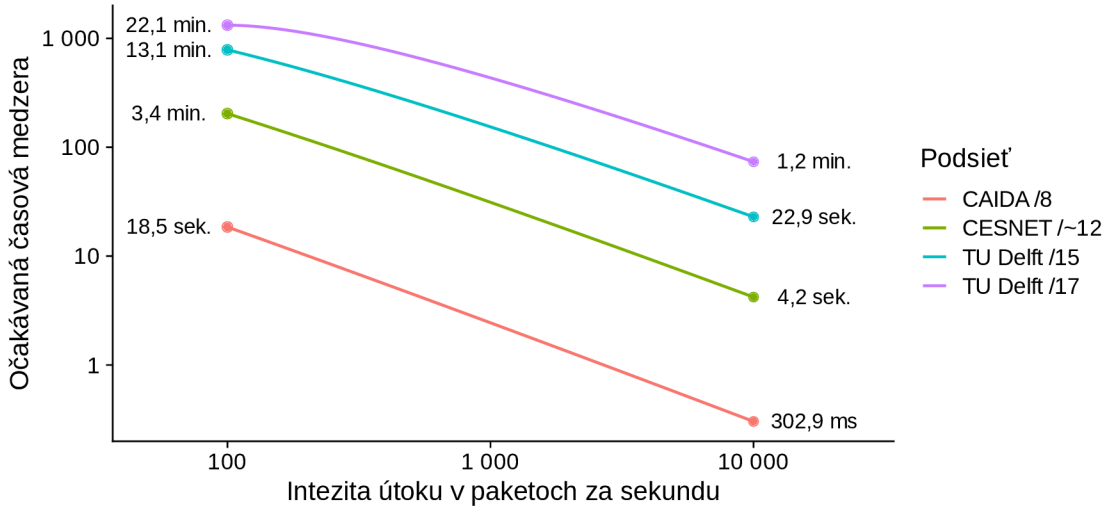


Obr. 4.3: Graf zobrazuje pravdepodobnosť pozorovania aspoň k spätne rozptýlených paketov v danej podsieti pri útoku s intenzitou 1 000 paketov za sekundu a trvaním tri minúty. Na obrázku sú taktiež uvedené ilustračné body s presnou hodnotou pravdepodobnosti a počtom paketov k.

4.4 Očakávaná časová medzera medzi paketmi

Pri detekcii útokov prúdovým spôsobom, je potrebné riešiť problém časovača. Teda určiť množstvo času od obdržania posledného paketu, kedy je možné považovať udalosť (útok) za skončenú. Príliš malá hodnota časovača vedie k rozdeleniu útoku na viac udalostí a príliš veľká hodnota môže viesť k spojeniu nezávislých útokov do jednej udalosti.

V tejto časti analýzy je stále predpokladom určitá konštantná intenzita útoku. Pakety takéhoto útoku budú generované so zhruba rovnakými časovými rozostupmi. Proces podvrhnutia IP adries však prebieha náhodne a teda časový rozdiel pozorovania medzi dvojicami spätne rozptýlených paketov na danej podsieti bude narozdiel od generovania útočných paketov nepravidelný. Pre určenie časovača je potrebné vypočítať teoretický počet po sebe idúcich paketov k v sekvencii všetkých paketov útoku N, ktorých zdrojová adresa je podvrhnutá tak, že na pozorovanú podsieť nedorazí žiadny paket zo sekvencie k. Očakávaná najdlhšia veľkosť tejto postupnosti pre podsieť s pravdepodobnosťou obdržania paketu p je daná vzorcom 4.9 [16, 36]. Pomer k a intenzity útoku potom určuje očakávanú časovú medzeru medzi pozorovanými paketmi na danej podsieti. Na obrázku 4.4 sú zobrazené rôzne hodnoty očakávanej časovej medzery v závislosti na intenzite útoku. Z obrázku je pozorovateľný pomerne veľký rozdiel časovej medzery vzhľadom na intenzitu útoku. Preto je voľba časovača kompromisom medzi rozdelením menej intenzívnych útokov do viacerých udalostí a spojením intenzívnejších útokov do jednej udalosti. Ďalším faktorom, ktorý ovplyvňuje voľbu časovača, je dĺžka trvania útoku. Pri dlhších útokoch s konštantnou intenzitou zasielania paketov bude očakávaná najdlhšia medzera prirodzene väčšia. Ak napríklad očakávame, že väčšina útokov nepresiahne intenzitu 100 paketov za sekundu a dĺžku jednej hodiny, potom je pre organizáciu CESNET rozumné nastaviť hodnotu časovača aspoň na 3,4 minúty. V prípade tohto útoku je očakávaný počet pozorovaných paketov na organizácii CESNET 77. Tento útok nie je moc veľký a teda aj počet pozorovaných paketov je pomerne malý.



Obr. 4.4: Očakávaná časová medzera v útoku s trvaním jedna hodina v závislosti od veľkosti podsiete a intenzite daného útoku.

Voľba menšej hodnoty časovača by však viedla k rozdeleniu takéhoto útoku na niekoľko nezávislých udalostí, čo ďalej komplikuje a znepresňuje prípadné odvodenie intenzity útoku.

$$k \ = log_{\frac{1}{1-p}} Np \tag{4.9}$$

Táto kapitola skúmala teoretické možnosti detekcie DDoS útokov s použitím spätného rozptylu. Uvedené teoretické poznatky boli demonštrované na niekoľkých príkladoch, ktoré ukazujú, že na veľkosti teleskopu záleží, konkrétne čím väčší teleskop k dispozícii máme tým je detekcia útokov rýchlejšia a presnejšia. Tento záver je pomerne dôležitý najmä v kontexte skutočného rozloženia výskytu útokov z hľadiska ich dĺžky a intenzity, ktorý zobrazujú obrázky 3.3 a 3.4. Z obrázkov je možné pozorovať, že na internete prevažujú najmä malé a krátke útoky, čo zdôrazňuje potrebu dostatočne veľkého teleskopu za účelom ich pozorovania.

Kapitola 5

Detekcia DDoS útokov založená na spätnom rozptyle

V tejto kapitole sú popísané existujúce prístupy k detekcii DDoS útokov skrz spätný rozptyl. Rozobrané prístupy vychádzajú ako z jednoduchých kvantitatívnych pravidiel na identifikovanie útokov, tak aj použitia strojového učenia. Jednotlivé metódy sú zoradené chronologicky podľa dátumu publikácie uvedených zdrojov.

Samotná detekcia prebieha najmä v prostredí tzv. teleskopov – nepriradených adresových blokov, kde nie sú žiadne legitímne zariadenia. Na teleskope sú spravidla len tri typy komunikácie: DDoS útoky, scany a komunikácia, ktorá vznikla chybami v konfigurácii sieťových prvkov [1]. Súvislosti teleskopov a spätného rozptylu sa podrobnejšie venuje sekcia 3.2.

V prípade klasifikátorov založených na strojom učení sú pre vyhodnotenie typicky použité metriky: citlivosť (*TPR – True positive rate*), špecifickosť (*TNR – True negative rate*) a prediktívna hodnota pozitívneho testu (*precison*). Uvedené metriky sú definované nasledovne:

$$\text{Citlivosť} = \frac{\text{Počet správne identifikovaných pozitívnych vzoriek}}{\text{Celkový počet pozitívnych vzoriek v sade}}$$

$$\text{Špecifickosť} = \frac{\text{Počet správne identifikovaných negatívnych vzoriek}}{\text{Celkový počet negatívnych vzoriek v sade}}$$

$$\text{Prediktívna hodnota pozitívneho testu} = \frac{\text{Počet správne identifikovaných pozitívnych vzoriek}}{\text{Celkový počet vzoriek, ktoré označila metóda ako pozitívne}}$$

Pozitívne vzorky predstavujú DDoS útok a negatívne všetko ostatné, teda hlavne scany a miskonfigurácie. Citlivosť a špecifickosť sa viažu k dátovej sade a hovoria, koľko pozitívnych/negatívnych vzoriek sa nám celkovo podarilo identifikovať. Pri detekcii DDoS útokov je ale pomerne dôležité, aby nedošlo k falošnému poplachu a klasifikátor neoznačil komunikáciu nesprávne ako DDoS útok. Na túto skutočnosť je citlivá posledná z uvedených metrík, tá vyjadruje aký pomer zo všetkých nahlásených DDoS udalostí skutočne predstavujú DDoS útoky. Citlivosť a prediktívna hodnota pozitívneho testu sa ďalej kombinuje pomocou harmonického priemeru do metriky označovanej F1 skóre. V článkoch sa typicky neuvádza presnosť, nakoľko sú dátové sady výrazne nevyvážené a preto sa používajú vyššie popísané metriky.

5.1 CAIDA

Organizácia CAIDA (*Center for Applied Internet Data Analysis*) má k dispozícii pomerne veľký teleskop s prefixom siete /8. Prostredníctvom tohto teleskopu monitorujú zastúpenie DDoS útokov na internete. Metóda identifikácie DDoS útokov, ktorú CAIDA vytvorila je podrobne popísaná v [17]. Táto metóda je na teleskope CAIDA implementovaná formou zásuvného modulu označeného ako `RS DoS`. Detaily implementácie tohto modulu sú dostupné online[1]. Výsledky modulu sú na dennej báze ukladané do DDoS dátovej sady s rovnomenným označením [4].

Samotný `RS DoS` modul vyžaduje funkčnosť ďalšieho CAIDA modulu `FlowTuple`. Modul `FlowTuple` agreguje pakety vo zvolenom časovom okne na základe zhody osem prvkovej hlavičky. Hlavička pozostáva zo zdrojovej IP adresy, cieľovej IP adresy, zdrojového portu, cieľového portu, protokolu, TTL, TCP príznakov a veľkosti IP datagramu. V prípade, ak sa jedná o protokol ICMP, zdrojový, resp. cieľový port reprezentuje ICMP typ, resp. kód. Agregovaná hodnota potom predstavuje počet výskytov paketov so zhodnou hlavičkou v danom časovom intervale. Jedná sa teda o určitú formu toku. Modul `FlowTuple` klasifikuje tieto toky do troch kategórií *Backscatter*, *ICMP Request* a *Other*, pričom pre modul `RS DoS` je podstatná trieda *Backscatter*, ktorá predstavuje toky so spätne rozptýlenými paketmi. K toku je priradená značka spätného rozptylu[2], ak sa jedná o TCP paket s príznakmi ACK-SYN alebo príznakom RST a ICMP s typmi 0, 3, 4, 5, 11, 12, 14, 18. Uvedené ICMP typy reprezentujú tie pakety, ktoré vznikli ako odpoveď na predošlú komunikáciu alebo indikujú výskyt chyby. Príkladom je typ 0 – odpoveď na požiadavku (*Echo reply*) alebo typ 3 – nedostupnosť cieľa (*Destination unreachable*).

Samotný modul `RS Dos` je postavaný na kvantitatívnych podmienkach, ktoré autori určili na základe štatistík vychádzajúcich z pozorovania spätne rozptýlených paketov na CAIDA teleskope. Aby bola IP adresa na základe spätne rozptýlených tokov z predošlého kroku (`FlowTuple`) klasifikovaná ako útok musí spĺňať tri podmienky:

1. Počet paketov je väčší ako 25
2. Útok trval dlhšie ako minútu
3. Útok dosiahol v aspoň jednom jednominútovom intervale tohto útoku intenzitu 0,5 paketov za sekundu, teda 30 paketov za minútu

Ďalším parametrom, ktorý ovplyvňuje výslednú klasifikáciu, je doba neaktívneho časovača (timeout). V predvolenom režime je táto doba 5-minút, teda ak počas 5-tich minút nie je od danej obete obdržaný žiadny paket, útok sa považuje za skončený. Autori taktiež k jednotlivým podmienkam uvádzajú príklady útokov vzhľadom na veľkosť ich teleskopu. Prvej podmienke napríklad odpovedá útok, ktorý trvá 22-minút s intenzitou 56,6 Kb/s pri paketoch s veľkosťou 1 500 bajtov. Tretia podmienka odpovedá útoku o rýchlosti 41 Kb/s v prípade 40 bajtových TCP paketov a 1.5 Mb/s v prípade 1 500 bajtových paketov. Prvá a tretia podmienka filtruje útoky s minimálnym dopadom. Druhá podmienka a neaktívny časovač predstavuje skôr voľbu, ktorá určuje aké minimálne trvanie majú útoky mať a kedy je možné považovať útok za skončený. Zvyšovaním neaktívneho časovača sa prirodzene zmenšuje počet útokov nahlásených `RS DoS` modulom, nižšie hodnoty budú naopak potenciálne viesť k rozdeleniu jednotlivých útokov na viacero udalostí.

[1] https://www.caida.org/tools/measurement/corsaro/docs/plugins.html#plugins_aggregation
[2] https://www.caida.org/tools/measurement/corsaro/docs/corsaro__flowtuple_8c_source.html#l00204

Výstup `RS Dos` modulu obsahuje 12 položiek: IP adresu obete, časovú značku začiatku útoku (prvý pozorovaný paket útoku), časovú značku konca útoku (posledný pozorovaný paket útoku), počet rozdielnych IP adries útočníkov, počet rozdielnych portov útočníkov, počet rozdielnych portov obetí, počet paketov útoku, počet bajtov útoku, maximálny počet paketov za minútu, krajinu IP adresy obete v čase útoku, kontinent pôvodu IP adresy obete v čase útoku a jej autonómny systém (ASN).

5.2 NICT

Až tri články [18, 19, 20] klasifikácie DDoS útokov skrz spätne rozptýlené pakety pochádzajú z Národného inštitútu informačných a komunikačných technológií v Japonsku (NICT[3]). V každom z uvedených článkov používajú pre rozpoznanie DDoS útokov algoritmy strojového učenia, ktoré boli trénované na dátach z teleskopu NICT, pričom použitá časť resp. veľkosť NICT teleskopu sa v jednotlivých článkoch líši. V prvom článku [18] veľkosť uvedená nie je. V druhom článku [19] uvádzajú veľkosť NICT teleskopu na 140 000 IP adries, čo je zhruba 2^{17} (teleskop /15) a v poslednom [20] použili teleskop s prefixom siete /16. Ďalej sa články líšia najmä rozdielnou metódou strojového učenia a dátovou sadou s ohľadom na metodiku jej tvorby, veľkosť a aktuálnosť.

Ideou použitia strojového učenia namiesto kvantitatívnych podmienok je natrénovanie klasifikátoru na ľahko rozlíšiteľnej spätno rozptýlenej prevádzke a následná klasifikácia horšie rozlíšiteľných útokov len skrz abstrahované rysy.

SVM

Prvý článok od NICT predstavuje [18]. Cieľom autorov je natrénovanie SVM modelu na TCP spätnom rozptyle z portu 80 a následné aplikovanie tohto modelu aj na iné porty a prípadne aj protokol UDP. Autori teda vychádzajú z predpokladu, že pri zmene portu či protokolu si rysy popisujúce DDoS útok zachovajú podobnú charakteristiku ako v prípade útoku na TCP/80. Kombinácia protokolu TCP a portu 80 bola v článku použitá preto, lebo sa asociuje s protokolom HTTP, ktorý je častým cieľom DDoS útokov.

Klasifikovanie prebieha na úrovni IP adresy zdroja paketov zasielaných na teleskop a IP adresa je označená ako obeť DDoS útoku, ak spĺňa dve podmienky:

1. Za mesiac odošle na teleskop viac ako 100 paketov
2. Každý z paketov obsahuje jednu z kombinácii príznakov: SYN-ACK, RST-ACK, RST alebo ACK

Udalosť sa považuje za skončenú, ak z danej IP adresy nepríde na teleskop žiadny ďalší paket v priebehu jednej hodiny. K IP adrese zdroja je následne vypočítaný vektor rysov. Vektor rysov je vypočítaný len z paketov určitého časového okna, ktorého počiatok je daný príchodom prvého paketu v rámci pozorovanej IP adresy zdroja. Vektor rysov je teda tvorený len z paketov zozbieraných zo začiatku komunikácie od daných zdrojov a pozostáva z nasledujúcich prvkov:

1. Počet paketov
2. Počet unikátnych zdrojových portov

[3] https://www.nict.go.jp/en/

3. Zastúpenie paketov so zdrojovým portom 80
4. Počet unikátnych cieľových IP adries
5. Priemerný počet paketov na unikátnu cieľovú IP adresu
6. Variancia počtu paketov na unikátnu cieľovú IP adresu
7. Počet unikátnych cieľových portov
8. Priemerný počet paketov na unikátny cieľový port
9. Variancia počtu paketov na unikátny cieľový port
10. Priemerná veľkosť obsahu paketu
11. Variancia veľkosti obsahu paketu

Tieto rysy sa v určitej obmene vyskytujú aj v nasledujúcich NICT článkoch, s výnimkou vlastnosti 3, ktorá sa už v ďalších článkoch nevyskytovala.

Trénovacia dátová sada bola zozbieraná z 20-tich dní a testovacia z 11-tich dní. Celkovo obsahujú dátové sady 1 985 IP adries označených ako DDoS pozitívne a 81 ako DDoS negatívne. Sady boli pred trénovaním a testovaním vyvážené metódou opakovania položiek (*oversampling*). Sady boli anotované len na základe TCP komunikácie na porte 80.

Na trénovanie a následnú klasifikáciu bol použitý model SVM s RBF jadrovou funkciou. V článku vyhodnotili niekoľko verzií klasifikátoru v závislosti na rôznej veľkosti okna, z ktorého je počítaný vektor rysov. Autori experimentovali s veľkosťou okna od 15-tich do 90-tich sekúnd s 15 sekundovými krokmi. Výsledky sa pre jednotlivé veľkosti okna moc nelíšili. Vo všetkých prípadoch veľkostí dosiahli klasifikátory hodnotu citlivosti a špecifickosti aspoň na úrovni 90 %. Konkrétne pre najväčšie 90 sekundové okno dosiahol klasifikátor hodnotu citlivosti 93 % a špecifickosti 98 %.

RAN-LHS

V druhom článku [19] autori okrem kombinácie komunikácie TCP/80 (HTTP) používajú na trénovanie taktiež UDP/53 (DNS). Cieľom tohto článku nie je len klasifikovanie DDoS útokov na iných kombináciách portov a protokolov, ale aj časová efektivita učenia a procesu klasifikácie.

Klasifikácia DDoS útokov je narozdiel od predošlého článku daná na úrovni IP adresy zdroja a aj času, teda trieda pre danú IP adresu sa v čase môže líšiť. IP adresa zdroja je pre určitý časový interval klasifikovaná ako DDoS útok, ak všetky pakety pochádzajú z TCP/80, respektíve UDP/53 a majú príznaky SYN-ACK alebo RST, respektíve sa musí jednať o DNS dotaz, kde doménové meno obsahuje jedno z kľúčových slov http, www, com. Zároveň musí byť počet paketov v prvej minúte útoku aspoň 40. Ak nie sú obe podmienky splnené, IP adresa je v danom časovom intervale označená ako DDoS negatívna. Hodnota neaktívneho časovača, kedy sa útok považuje za skončený, je opäť 1 hodina.

Následné predspracovanie zvolených paketov a určenie rysov je takmer identické ako v [18]. Autori pri extrahovaní používajú 60 sekundové okno. Voľba vlastností sa od predošlého článku mierne líši, bola odstránená vlastnosť zastúpenia portu 80 a bolo pridaných 7 nových vlastností:

1. Priemerný časový rozdiel po sebe idúcich paketov
2. Variancia časového rozdielu po sebe idúcich paketov
3. Priemerný počet paketov na unikátny zdrojový port
4. Variancia počtu paketov na unikátny zdrojový port
5. Počet protokolov
6. Priemerný rozdiel cieľových IP adries v po sebe idúcich paketoch
7. Variancia rozdielu cieľových IP adries v po sebe idúcich paketoch

Na trénovanie autori použili 3 rôzne modely RBFNN, RAN a RAN-LHS. Posledné dva modely predstavujú vylepšenie neurónovej siete s radiálnymi bázovými funkciami (RBFNN), tak aby bolo možné túto sieť učiť inkrementálne. Všetky tri modely dosiahli veľmi podobné výsledky z pohľadu metrík klasifikácie. Najrýchlejším modelom z hľadiska učenia a procesu klasifikácie bol model RAN-LHS s nasledovnými hodnotami metrík: citlivosť 97,8 %, prediktívna hodnota pozitívneho testu 97,4 % a F1 skóre 97,5 %.

Trénovacia dátová sada bola zbieraná počas obdobia jedného roku a obsahuje 9 968 položiek, z toho 9 404 DDoS pozitívnych a 564 DDoS negatívnych. Testovacia dátová zachytáva obdobie 20-tich dní a obsahuje 5 933 položiek, kde 2 464 je pozitívnych a 3 469 negatívnych. V článku nie je jasne formulované ako bola testovacia sada získaná, ale pravdepodobne sa jedná len o pakety z kombinácií TCP/80 a UDP/53.

Zhlukovanie

V poslednom článku [20] zo sekcie NICT je použitý algoritmus vyvíjajúceho Cauchyho posibilistického zhlukovania (*evolving Cauchy possibilistic clustering*). Hlavným rozdielom oproti predošlým metódam je použitie algoritmu zhlukovania, to ale zároveň znamená, že pre klasifikačné účely je potrebné zhluky anotovať. Použitý algoritmus je navrhnutý na prúdové spracovanie a priebežné aktualizovanie zhlukov počas jeho činnosti. Z toho vyplýva prípadná potreba kategorizovania nových zhlukov aj po jeho nasadení.

Postup prevodu paketov na vektor vlastností je takmer identický ako v [19] a voľba rysov je totožná. Líši sa len dĺžka intervalu, z ktorého je počítaný vektor rysov. Tento interval má na začiatku veľkosť 30 sekúnd a rozširuje sa až kým neobsahuje aspoň 20 paketov útoku. Hodnota neaktívneho časovača je opäť jedna hodina. Autori sa však vyjadrujú pomerne stručne ohľadom spôsobu identifikovania DDoS pozitívnych a negatívnych položiek v dátovej sade. Uvádzajú len, že pre identifikovanie použili známe postupy.

Trénovacia dátová sada pozostáva z 10-tich dní a obsahuje 8 819/407 TCP/UDP DDoS pozitívnych vektorov a 31 911/84 367 TCP/UDP DDoS negatívnych vektorov. Testovacia sada bola zozbieraná počas 50-tich dní a pozostáva z 33 217/2 077 TCP/UDP DDoS pozitívnych vektorov a 177 794/410 020 TCP/UDP DDoS negatívnych vektorov.

Autori vyhodnotili metódu osobitne pre protokol TCP a UDP. Metóda dosahuje na protokole TCP hodnotu citlivosti 98.1 %, prediktívnej hodnoty pozitívneho testu 98.2 % a F1 skóre 98.2 %. Na protokole UDP je citlivosť 75.8 %, prediktívna hodnota pozitívneho testu 70.1 % a F1 skóre 72.8 %.

5.3 TU Delft

V tomto článku [1] boli identifikované spätne rozptýlené pakety na teleskope Technickej univerzity v Delft. Identifikovanie prebehlo za účelom štatistického vyhodnotenia rozsahu a čestnosti DDoS útokov na internete. Autori preto stanovili niekoľko pravidiel na identifikovanie spätne rozptýlených paketov pre protokoly TCP, UDP a aj ICMP.

Za TCP rozptýlené pakety považujú tie s príznakmi SYN-ACK alebo RST. Pri protokole ICMP pozorovali autori najmä dva typy ICMP 0 – 70 % (echo) a 3 – 24 % (nedostupnosť cieľa). Prvý z uvedených typov prisúdili scanom a druhý spätne rozptýleným paketom. V prípade protokolu UDP autori uvádzajú nutnosť spracovania protokolov na vyššej úrovní s obmedzením sa na často používané a štandardizované porty. Bližšie informácie k spracovaniu UDP však neposkytujú. Ako príklad extrahovania informácií uvádzajú protokol DNS a zistenie znalosti o tom, či sa jedná o odpoveď alebo žiadosť.

Kapitola 6

Návrh detekcie DDoS útokov zo spätného rozptylu v dátach sieťových tokov

V predošlej kapitole bolo prezentovaných niekoľko existujúcich prístupov k detekcii DDoS útokov prostredníctvom spätného rozptylu. Všetky z uvedených metód vyžadujú na fungovanie tzv. sieťový teleskop. Jedná sa o pomerne veľký nepriradený a monitorovaný adresový priestor s absenciou legitímnej komunikácie. Komunikácia smerujúca na adresový rozsah teleskopu je typicky pôvodom len z troch aktivít: DDoS útokov (spätný rozptyl), scanov a miskonfigurácií. To robí detekciu DDoS útokov v prostredí teleskopu pomerne priamočiaru a je možné ju založiť len na kvantitatívnych prahoch a príznakoch paketov ako v prípade organizácie CAIDA [17]. Táto práca si kladie za cieľ vytvoriť metódu, ktorá bude schopná detegovať DDoS útoky na základe spätného rozptylu aj mimo teleskopov v prostredí legitímnej komunikácie a to navyše z dát sieťových tokov. V tejto kapitole je rozobratý návrh takého systému. Obsah kapitoly je rozdelený do dvoch častí. Prvá sa zaoberá rozdielom v spôsobe identifikovania spätne rozptýlených paketov v NetFlow dátach voči teleskopu a druhá predstavuje návrh riešenia.

6.1 Problémy identifikovania spätného rozptylu v prostredí NetFlow

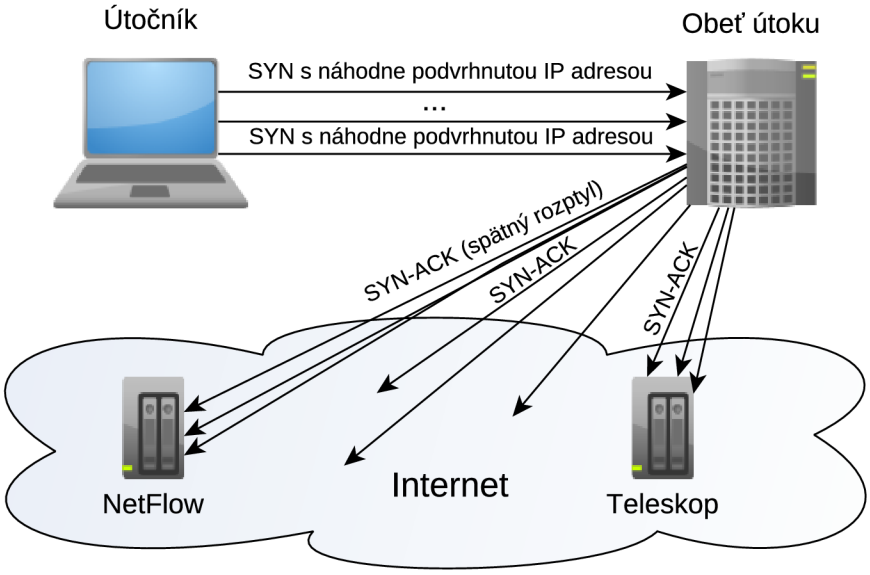
Prítomnosť legitímnej komunikácie v dátach robí detekciu rozptýlených paketov obtiažnejšiu a jednoduché použitie kvantitatívnych pravidiel v kombináciami s príznakmi preto nie je v takomto prostredí aplikovateľné tak, ako bolo prezentované v existujúcich prístupoch v kapitole 5. Napríklad niekoľko TCP paketov s príznakmi SYN-ACK je v prostredí teleskopu možné pripísať DDoS útoku (jedná sa o spätne rozptýlené pakety) pretože jediný iný spôsob akým tieto pakety mohli vzniknúť je vplyvom chybnej konfigurácie siete. Oddelenie od miskonfigurácií je potom možné učiniť len na základe kvantitatívnej podmienky. Pridaním legitimnej komunikácie do monitorovacej siete sa situácia komplikuje a TCP pakety s príznakmi SYN-ACK mohli vzniknúť aj ako dôsledok oprávnených požiadavkov niekoľkých klientov voči danej službe. Z toho vypláva potreba zaviesť dotačné heuristiky a prípadne zvýšiť kvantitatívne prahy, aby bolo možné odlíšiť legitímnu komunikáciu a útok.

Detekciu útokov zo spätného rozptylu v NetFlow ďalej komplikuje samotná reprezentácia dát v podobe sieťových tokov. Toky samé o sebe neobsahujú informácie o obsahu

paketov a metóda je limitovaní len na informácie v tokoch, ktoré pochádzajú najmä z hlavičky paketu. Informácie z hlavičiek sú navyše v agregovanej podobe. Dôsledkom agregácie je, že v niektorých prípadoch nie je možné jednoznačne odvodiť obsah hlavičiek jednotlivých paketov. Uvažujme situáciu z obrázku 3.1, kde je zobrazené uzatvorenie TCP spojenia (*three-way handshake*) medzi klientom a serverom. Táto komunikácia by bola v prostredí NetFlow reprezentovaná dvomi tokmi jedeným od klienta a druhým od serveru. Z obrázku je zrejmé, že nie len tok zo strany serveru, ale aj tok od klienta k serveru bude mať príznaky SYN-ACK nakoľko dôjde k zlúčeniu príznakov ACK a SYN v paketoch od klienta. Tento problém môže prirodzene nastať len vtedy, ak tok obsahuje viac ako jeden paket. Preto je možné uvažovať vymedzenie analýzy len na jedno-paketové toky, aby bolo možné správne určiť pakety s príznakmi SYN-ACK. Toto obmedzenie však nie je ideálnym riešením, pretože obeť DDoS útoku môže po neúspešnom pokuse o spojenie spôsobené útočníkom zaslať po určitom čase paket s príznakmi SYN-ACK znova, jedná sa tzv. retransmisie, ktoré sa prejavia väčším počtom paketov v toku od serveru. Zanedbanie týchto tokov by potenciálne mohlo viesť k znemožneniu detekcie DDoS útokov na servery, ktoré používajú retransmisie.

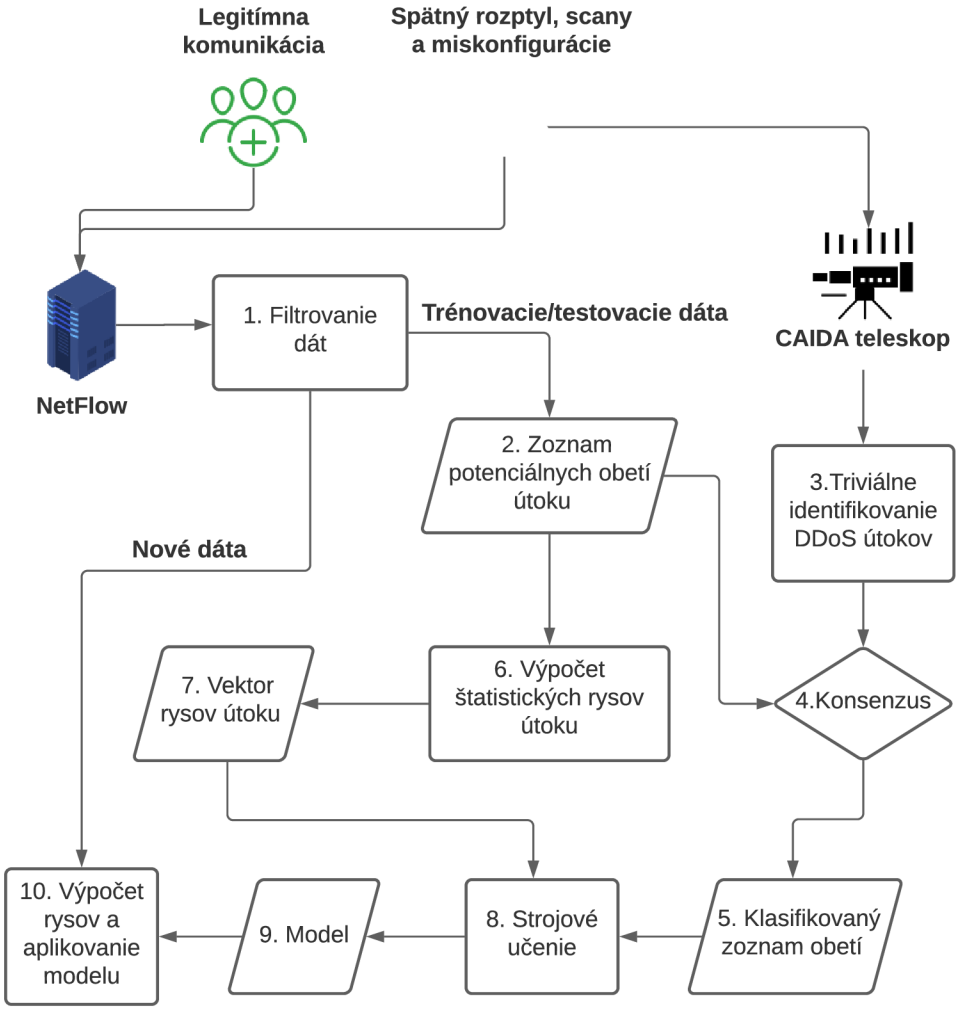
6.2 Návrh detekcie

Ako bolo uvedené v predošlej časti, detekcia rozptýlených paketov je v prostredí legitímnej komunikácie v spojení s dátami tokov pomerne problematická. Je však možné využiť náhodnosti podvrhnutia zdrojových IP adries v paketoch útoku, ktorá bola podrobnejšie diskutovaná v kapitole 4. Náhodné podvrhnutie teoreticky umožňuje sledovanie DDoS útokov z ľubovoľnej časti internetu. Je teda možné sledovať útok na teleskope, kde je identifikácia jednoduchšia a následne korelovať súčasný výskyt komunikácie v dátach tokov. Situácia je ilustrovaná na príklade TCP zahltenia v obrázku 6.1.



Obr. 6.1: Útok typu SYN zahltenie. Spätne rozptýlené pakety s náhodne podvrhnutými IP adresami smerujú ako na teleskop, tak do siete, kde je umiestnená NetFlow sonda.

Hlavná myšlienka návrhu spočíva v označení dát tokov pomocou teleskopu a následnou tvorbou modelu pomocou strojového učenia, ktorý bude schopný klasifikovať dáta tokov aj bez nutnosti použitia dát teleskopu. V tejto práci bol na identifikovanie DDoS útokov použitý teleskop organizácie CAIDA, ktorý je na požiadanie prístupný pre potreby výskumu. Tento teleskop je pomerne rozsiahly a je tvorený 16 777 216 IP adresami (prefix siete /8) [5]. Zdrojom dát tokov bola organizácia CESNET, ktorá je rozsahom zhruba 18-krát menšia ako teleskop organizácie CAIDA. Tento veľkostný rozdiel zdôrazňuje výhodu použitia teleskopu CAIDA na anotačné účely, nakoľko útok pozorovateľný v rozsahu CESNETu bude s veľkou pravdepodobnosťou pozorovaní na teleskope CAIDA, kde by malo navyše doraziť približné 18-krát viac paketov ako v prípade rozsahu CESNETu.



Obr. 6.2: Proces tvorby modelu detekcie DDoS útokov zo spätného rozptylu s použitím NetFlow dát. Číslovanie určuje postupnosť jednotlivých krokov.

Princíp anotácie a tvorby modelu je zobrazený na obrázku 6.2. Prvým krokom je filtrovanie dát tokov podobne ako filtrovanie paketov v metóde organizácie CAIDA. V tomto kroku sú ponechané len toky, ktoré môžu predstavovať spätný rozptyl. Jedná sa o TCP toky s príznakmi SYN-ACK alebo RST a ICMP pakety s typmi predstavujúcimi odpoveď respektíve reakciu na predošlú komunikáciu: 3, 4, 5, 11, 12, 14, 18. UDP pakety, resp. útoky

nie sú v tejto práci uvažované vzhľadom na menšie zastúpenie oproti TCP útokom a nutnosť analýzy obsahu UDP paketov. Útoky na protokole UDP sú však v určitej miere zastúpené v rámci ICMP komunikácie, nakoľko pri niektorých typoch UDP zahltenia dochádza ku generovaniu ICMP paketov s typom 3 od obete útoku [1, 17]. Ďalej sú v kroku filtrovania odstránené toky, ku ktorým neexistuje obojsmerná komunikácia. Nakoľko protokol TCP vyžaduje ustanovenie spojenia a uvedené ICMP typy predstavujú odpoveď, potom neexistencia obojsmernej komunikácie naznačuje podvrhnutie paketu, respektíve chybu v konfigurácii. Ak obojsmerná komunikácia existuje, potom sa s veľmi vysokou pravdepodobnosťou nejedná o náhodne podvrhnutý paket. Odstránenie obojsmernej komunikácie teda redukuje počet udalostí, ktoré je nutné v neskorších krokoch klasifikovať a navyše zjednodušuje sledovanie DDoS útokov, pretože určitým spôsobom simuluje prostredie teleskopu. Napríklad pri ponechaní obojsmernej komunikácie nie je možné vo všetkých prípadoch stanoviť počiatok a koniec útoku. Môže sa totiž stať, že okrem útočníka sú v sledovanom rozsahu súčasne prítomné legitímne zariadenia používajúce službu, ktorá ja obeťou útoku. V takom prípade dôjde k zmiešaniu legitimných a útočných paketov. Určenie počiatku a konca takéhoto útoku je potom možné len s použitím určitej heuristiky, napríklad sledovaním zmeny intenzity zasielaných paketov.

V druhom kroku je vytvorený zoznam potenciálnych obetí, jedná sa o IP adresy, od ktorých smeruje spätný rozptyl do monitorovanej siete prostredníctvom NetFlow. Podobný zoznam sa vytvorí aj v rámci teleskopu, a ak je IP adresa v rovnakom časovom intervale pozorovaná súčasne na teleskope a aj v NetFlow dátach, potom je potenciálna obeť klasifikovaná ako skutočná obeť (prebieha útok) a zvyšné IP adresy sú označené ako falošné obete (neprebieha útok). Tento bod je na obrázku 6.2 označený ako 4. krok – konsenzus. Pre každú potenciálnu obeť sú následne z dát tokov vypočítané rysy, ktoré reprezentujú útok na obeť, respektíve spätný rozptyl od obete. Klasifikovaný zoznam obetí spolu s rysmi sa potom použije na natrénovanie modelu strojového učenia (model učenia s učiteľom). Takto natrénovaný model bude ďalej k detekcii DDoS útokov potrebovať už len dáta tokov. Nakoľko sú spracovávané dáta pre protokol TCP ako aj ICMP, budú vo výsledku vytvorené dva modely. Jeden pre klasifikovanie TCP DDoS útokov a druhý pre ICMP DDoS útoky.

Rysy, ktoré reprezentujú útok boli prevzaté z metód [19, 20, 17] popísaných v kapitole 5. Narozdiel od uvedených metód, ktoré pracujú nad paketmi, boli rysy v tejto práci prispôsobené a rozšírené o ďalšie rysy aplikovateľné v prostredí sieťových tokov. Kompletný zoznam rysov je uvedený v tabuľke 6.1. V tejto práci bol kladený dôraz na výber tých rysov, ktoré je možné implementovať s minimálnymi pamäťovými a výpočtovými nárokmi.

V predošlom texte nebolo zámerne špecifikovaných niekoľko detailov návrhu, ktoré budú predmetom experimentov a implementácie. Jedná sa napríklad o voľbu strojového učenia. Ako vhodné sa javia klasifikátory založené na rozhodovacích stromoch, tie sú odolné voči nevyváženým dátovým sadám a taktiež poskytujú odhad správnosti klasifikácie, ktorý je možné použiť na nastavenie vhodného klasifikačného prahu vzhľadom na požadovanú citlivosť či špecifickosť.

Rys	Protokol	Zdroj/inšpirácia
Celkový počet bajtov	TCP, ICMP	
Počet paketov	TCP, ICMP	[19, 20]
Priemerný počet bajtov na paket	TCP, ICMP	[19, 20]
Štandardná odchýlka počtu bajtov na paket	TCP, ICMP	[19, 20]
Počet tokov	TCP, ICMP	
Priemerný počet paketov na tok	TCP, ICMP	
Štandardná odchýlka počtu paketov na tok	TCP, ICMP	
Maximálny počet tokov za minútu	TCP, ICMP	[17]
Priemerný počet tokov za sekundu	TCP, ICMP	
Počet unikátnych cieľových IP adries	TCP, ICMP	[19, 20]
Počet unikátnych cieľových podsietí s prefixom siete /24	TCP, ICMP	
Počet unikátnych cieľových portov	TCP	[19, 20]
Počet unikátnych zdrojových portov	TCP	[19, 20]
Počet unikátnych cieľových IP adries normalizovaný počtom tokov	TCP, ICMP	
Počet unikátnych cieľových podsietí s prefixom siete /24 normalizovaný počtom tokov	TCP, ICMP	
Počet unikátnych cieľových portov normalizovaný počtom tokov	TCP	
Počet unikátnych zdrojových portov normalizovaný počtom tokov	TCP	

Tabuľka 6.1: Zoznam všetkých rysov použitých v tejto práci. Pri každom ryse je uvedená možnosť jeho aplikovateľnosti na protokoloch TCP a ICMP. Tabuľka ďalej uvádza, z akého zdroja bol rys prevzatý, rysy bez zdroja boli navrhnuté v tejto práci. Uvedené rysy charakterizujú množinu spätne rozptýlených paketov pre každú obeť útoku. V prípade vyššie uvedených rysov určuje cieľová IP adresa príjemcu spätne rozptýlených paketov a nie cieľ útoku.

Kapitola 7

Implementácia navrhnutej metódy detekcie DDoS útokov

Táto kapitola sa detailnejšie venuje dôležitým implementačným prvkom. Proces detekcie DDoS útokov zo spätného rozptylu ako aj samotná implementácia je rozdelená do dvoch hlavných častí. Prvú časť tvorí program na extrakciu rysov zo spätne rozptýlených paketov. Rysy sú následne zaslané do druhej časti – klasifikačného programu. Pri klasifikácia dochádza k zaradeniu vektoru rysov do DDoS pozitívnej, respektíve negatívnej triedy. Uvedené rozdelenie úloh vyplýva najmä z požiadavky spracovania tokov v reálnom čase. Výpočtovo najzložitejšou časťou je práve výpočet rysov a preto je prvá časť implementovaná v jazyku C++. Klasifikácia naopak nie je kritická z hľadiska efektivity nakoľko spracováva len extrahované rysy, ktorých je výrazne menej ako tokov. Klasifikačný program ale vyžaduje možnosť flexibilnej zmeny a vývoja modelu klasifikácie. Z tohto dôvodu bol pre implementáciu druhej časti zvolený jazyk Python3, ktorý poskytuje množstvo vysokoúrovňových knižníc pre tvorbu modelov s použitím strojového učenia. Obe uvedené časti sú implementované formou modulov v rámci systému NEMEA [7].

7.1 NEMEA

NEMEA [7] je open source[1] modulárny systém vyvíjaný organizáciou CESNET[2], ktorý slúži na detekciu a analýzu prúdových dát sieťových tokov. Systém NEMEA je tvorený z troch hlavných častí: modulov, NEMEA frameworku a tzv. supervízora. Moduly predstavujú samostatné prvky systému NEMEA. Každý modul má určitú úlohu, resp činnosť, ktorú vykonáva. Jedná sa napríklad o spracovanie, filtrovanie či uloženie dát. V prípade, ak modul deteguje nejakú hrozbu označuje sa v rámci terminológie systému NEMEA ako detektor, pre zvyšné moduly sa používa len označenie modul. NEMEA aktuálne obsahuje desiatky dostupných modulov a detektorov, ktorých počet stále rastie. Medzi podporované jazyky tvorby modulov patrí C, C++ a Python.

NEMEA framework zabezpečuje jednotný spôsob komunikácie medzi modulmi tak, aby ich bolo možné jednoducho kombinovať. Základ komunikácie predstavujú knižnice TRAP a UniRec. Knižnica TRAP implementuje rozhranie zasielania správ a UniRec definuje dátový formát správ. Formát UniRec je podobný dátovej štruktúre jazyka C navyše však umožňuje dynamicky definovať formát správy za behu programu. Knižnice TRAP a UniRec boli

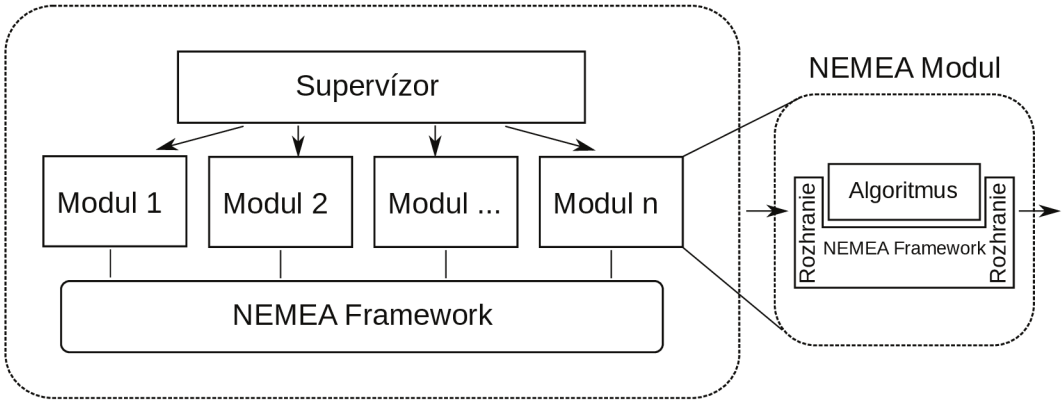
[1] https://github.com/CESNET/Nemea
[2] https://www.cesnet.cz/

vytvorené s dôrazom na vysokú efektivitu prenosu správ medzi modulmi nakoľko je systém NEMEA určený najmä na prácu s prúdovými dátami v reálnom čase.

Knižnica TRAP poskytuje tri typy komunikačných rozhraní: TCP soket, UNIXový soket a súborové rozhranie. Komunikácia pomocou protokolu TCP umožňuje spojenie modulov bežiacich na rôznych sieťových zariadeniach, UNIXový soket je určený na komunikáciu v rámci jedného zariadenia a súborové rozhranie poskytuje možnosť dáta uložiť a následne s nimi opakovane pracovať, napríklad pre účely testovania modulu. Všetky uvedené typy rozhrania je navyše možné kombinovať. Okrem knižníc TRAP a UniRec je súčasťou frameworku NEMEA aj knižnica Common, ktorá obsahuje sadu typicky používaných algoritmov a štruktúr.

Poslednú súčasť systému NEMEA predstavuje supervízor. Supervízor reprezentuje proces, ktorý dohliada na správny beh inštancie systému NEMEA. Vstupom supervízora je konfiguračný súbor vo formáte XML, v ktorom je definované prepojenie modulov. Supervízor potom na základe zvolenej konfigurácie spustí, respektíve ukončí dané moduly, priebežne kontroluje ich činnosť a vytvára rôzne štatistiky o nimi využívaných zdrojoch. Použitie supervízora v rámci systému NEMEA nie je nutné a moduly je možné spustiť aj individuálne. V prípade väčšieho počtu modulov však supervízor uľahčuje ich správu a nasadenie.

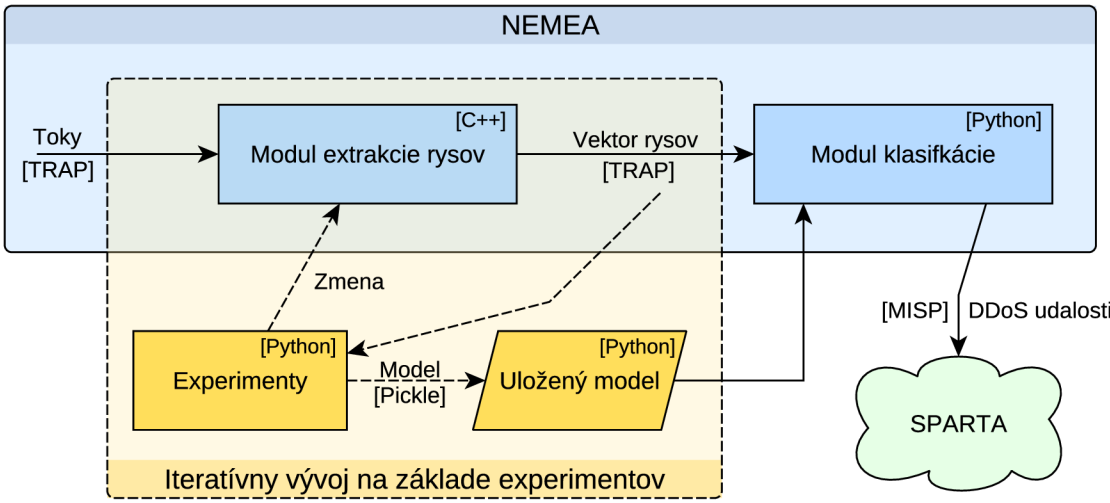
Obrázok 7.1 zobrazuje obecné schéma systému NEMEA. Daná schéma zachytáva hlavné úlohy systému NEMEA v podobe supervízora sledujúceho beh modulov a NEMEA frameworku, ktorý umožňuje predovšetkým rýchlu komunikáciu medzi modulmi. NEMEA teda definuje spôsob interakcie medzi modulmi a v réžii programátora zostáva len tvorba algoritmu modulu a konfiguračného súboru pre supervízora.



Obr. 7.1: Obecné schéma systému NEMEA [7], supervízor dohliada nad behom skupiny modulov, ktoré komunikujú pomocou frameworku NEMEA.

Na obrázku 7.2 je zobrazený proces vývoja a zapojenie modulu extrakcie rysov a modulu klasifikácie v rámci systému NEMEA. Základnú súčasť predstavuje modul extrakcie rysov, ktorý spolu s experimentmi predstavuje najzložitejšiu časť z hľadiska implementácie a návrhu. Modul extrakcie rysov bol vyvíjaný v niekoľkých iteráciách a to na základe odozvy z experimentov. Iteratívny vývoj je na obrázku 7.2 znázornený prerušovanou čiarou. V prvom kroku vývoja sú spočítané rysy z dát tokov, ktoré potom putujú do časti experimentov, kde je natrénovaný a vyhodnotený model strojového učenia. Na základe výsledkov modelu potom dochádza k prípadnej úprave modulu extrakcie, napríklad vo forme pridania nového rysu. Výsledkom iteratívneho vývoja je finálna implementácia modulu extrakcie rysov a predovšetkým model strojového učenia. Model je následne použitý v module klasifikácie, ktorá už nie je súčasťou iteratívneho vývoja, nakoľko funkcia tohto modulu spočíva len

v načítaniu modelu a vykonaní klasifikácie. V prípade, ak modul klasifikácie usúdi, že obdržaní vektor rysov pochádza z DDoS útoku dôjde k nahláseniu tejto udalosti do systému SPARTA[3] pomocou rozhrania MISP[4].



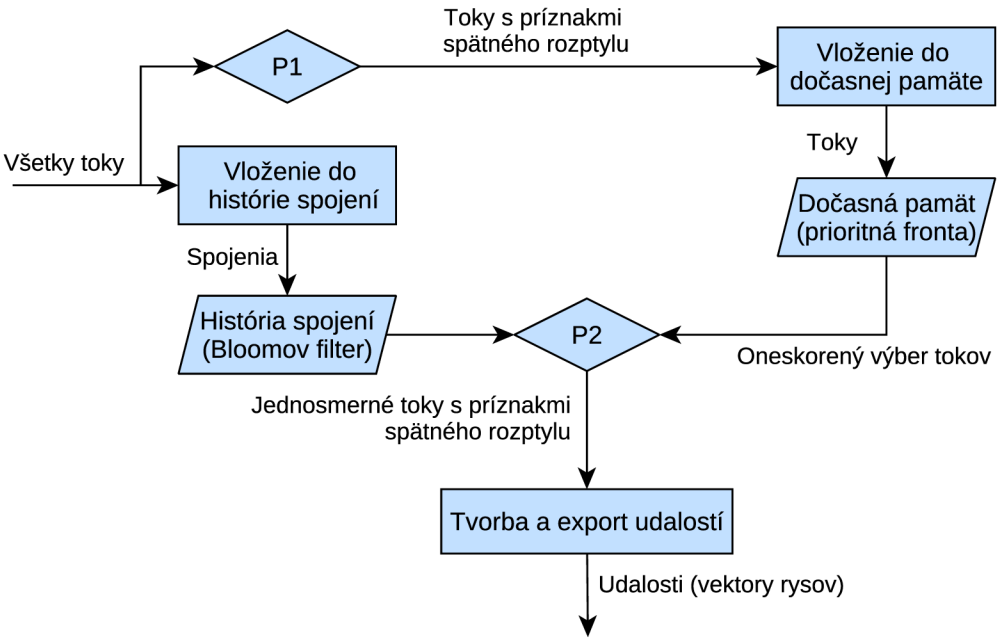
Obr. 7.2: Schéma prepojenia modulov a proces vývoja detektoru DDoS útokov zo spätného rozptylu v rámci systému NEMEA. Modrou farbou sú vyhradené časti systému NEMEA, žltá ohraničuje súčasti iteratívneho vývoja a zelená predstavuje systém SPARTA, do ktorého sú detegované útoky nahlásené. Prerušovanou čiarou sú naznačené dočasné závislosti platné v čase vývoja. Hranaté zátvorky uvádzajú použité nástroje.

7.2 Modul extrakcie rysov

Hlavnou úlohou modulu extrakcie rysov je sledovanie a vytvorenie reprezentácie jednosmernej komunikácie, ktorá potenciálne obsahuje spätne rozptýlené pakety, respektíve toky.

Základným predpokladom funkčnosti modulu je odfiltrovanie obojsmernej komunikácie, teda komunikácie, kde je možné pozorovať zaslanie paketov od oboch komunikujúcich zariadení. Obojsmernosť je možné overiť na základe uchovania určitého počtu tokov v dočasnej pamäti. Z tejto pamäte sú potom postupne vyberané jednotlivé toky a pri ich vybraní dochádza k vyhľadaniu opačného smeru. V tejto práci je však potrebné spracovať len toky s príznakmi spätného rozptylu, uchovávanie všetkých tokov je preto zbytočné. Zvyšné toky sú totiž použité len za účelom určenia existencie opačného smeru k tokom s príznakmi spätného rozptylu, ich samotný obsah v podobne nekľúčových položiek toku nie je nutné uchovávať. Z tohto dôvodu nie je pre určenie obojsmernosti použitá jedna dočasná pamäť ale dve. Prvá z nich je ďalej v texte označovaná ako história spojení a uchováva len informáciu o spojení – dvojici zdrojovej a cieľovej IP adresy daného toku. Druhá sa označuje ako dočasná pamäť a obsahuje kompletné toky. Zavedením dvoch pamätí dôjde k výraznému zníženiu pamäťových nárokov modulu, nakoľko toky s príznakmi spätného rozptylu reprezentujú len zlomok zo všetkých tokov.

[3] https://www.sparta.eu/
[4] https://www.misp-project.org/



Obr. 7.3: Diagram zobrazujúci spracovanie dát tokov modulom extrakcie rysov. P1 predstavuje podmienku na odstránenie tokov bez príznakov spätného rozptylu a P2 podmienku odstránenia obojsmerných tokov, respektíve tokov bez spojenia v oboch smeroch. Spojenie reprezentuje dvojicu zdrojovej a cieľovej IP adresy.

Proces spracovania tokov zobrazuje diagram na obrázku 7.3. Prvým krokom spracovania je uloženie spojenia do histórie spojení. Po uložení informácie o spojení smerujú toky s príznakmi spätného rozptylu do dočasnej pamäte. Toky sú z dočasnej pamäte postupne odoberané, pričom pri odobraní dochádza k vyhľadaniu opačného smeru spojenia daného toku v histórii spojení, ak opačné spojenie v histórii existuje značí to, že stanice spolu komunikovali v oboch smeroch a teda sa pravdepodobne jedná o legitímnu komunikáciu. Aby mohlo dôjsť k správnemu určeniu vzájomnosti komunikácie je vyberanie tokov z dočasnej pamäte oproti histórii časovo oneskorené, pri okamžitom overení by totiž história nemusela obsahovať oba smery komunikácie vzhľadom na rôznu odozvu zariadení, smerovanie paketov a čas exportu príslušných tokov. História spojení spolu s dočasnou pamäťou v podstate simuluje prostredie teleskopu, pretože do fázy spracovania rysov prepustí len komunikáciu, ku ktorej neexistuje odpoveď, čo naznačuje jej potenciálne podvrhnutie.

História však nie je dokonalá, nakoľko je za účelom vyššej efektivity implementovaná pomocou heuristiky v podobe Bloomovho filtra [2]. Ďalším faktorom ovplyvňujúcim správnosť určenia vzájomnej komunikácie je dĺžka zotrvania položky v dočasnej pamäti, prirodzene, čím je táto doba kratšia, tým viac sa zvyšuje pravdepodobnosť chybného určenia vzájomnosti spojenia.

Po odfiltrovaní obojsmernej komunikácie smerujú zostávajúce toky do fáze tvorby udalostí. Udalosť je významovo takmer identická s vektorom rysov, s tým rozdielom, že udalosť je dvojica pozostávajúca z vektoru rysov a identifikátoru, ktorý jednoznačne identifikuje potenciálny útok v tzv. tabuľke udalostí. Na základe identifikátoru udalosti je práve spracovávaný tok priradený k odpovedajúcej udalosti alebo je vytvorená nová udalosť, ak ešte

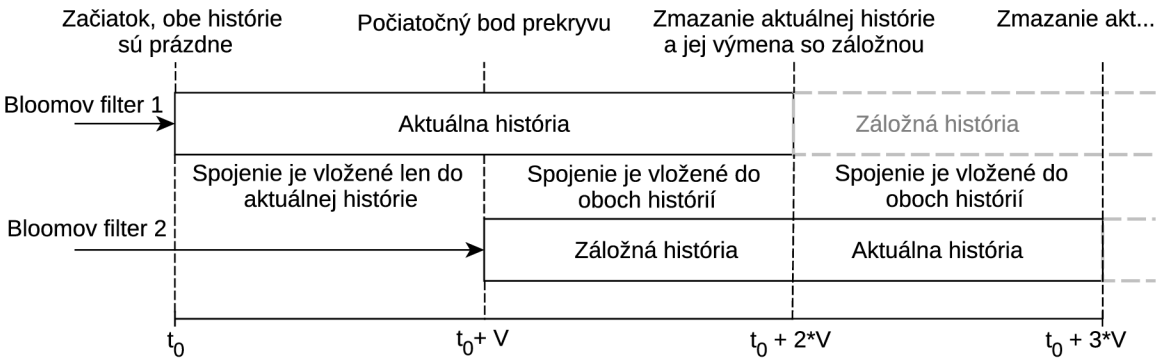
neexistuje. Pridanie toku do udalosti znamená spracovanie jednotlivých položiek toku, na základe ktorých sú vypočítané štatistické rysy reprezentujúce potenciálny útok. Udalosť je potom spolu s jej rysmi exportovaná do modulu klasifikácie. K exportu môže dôjsť po uplynutí pasívneho či aktívneho časovača podobne ako pri exporte tokov na kolektore.

História spojení a dočasná pamäť

História spojení je reprezentovaná heuristikou v podobe Bloomovho filtru [2], jedná sa o pravdepodobnostnú štruktúru, ktorú je možné použiť na zistenie prítomnosti prvku v množine s dôrazom na nízke pamäťové nároky. Bloomov filter je náchylný k falošne pozitívnym chybám, ale nemôže dôjsť k falošne negatívnej chybe. To znamená, že ak prvok do množný patrí, potom bude odpoveď Bloomovho filtra vždy správna, ak sa ale prvok v množine nenachádza v niektorých prípadoch môže dôjsť k nesprávnej odpovedi, teda tej, že sa prvok v množine nachádza, aj keď tam v skutočnosti nie je. Očakávaná miera tejto chybovosti, respektíve miera falošne pozitívnych prípadov je nastaviteľným parametrom Blomovho filtra. Základnými prvkami Bloomovho filtra je úložisko v podobe bitovej tabuľky a hašovacie funkcie použité na adresovanie buniek tejto tabuľky. Bity tabuľky sú na začiatku inicializované na nulu. Vloženie prvku do tabuľky potom vyžaduje výpočet niekoľkých hašovacích funkcií, ktoré určia adresy kde bude zapísaní bitová hodnota 1. Pri overení príslušnosti prvku v množine sa výpočet zopakuje, a ak je na každej adrese hodnota jedna, potom podľa Bloomovho filtra prvok do množiny patrí. Počet hašovacích funkcií závisí od prípustnej chybovosti, čím menšia chybovosť je požadovaná, tým viac je potrebných hašovacích funkcií a teda rastú aj výpočtové nároky [2]. V tejto práci bola použitá voľne dostupná implementácia Bloomovho filtra[5], ktorá je taktiež súčasťou frameworku NEMEA[6]. Uvedená knižnica do veľkej miery abstrahuje prácu s Bloomovým filtrom a pre jeho použite je potrebné špecifikovať len očakávaný počet vložených položiek a prípustnú chybovosť. Hašovacie funkcie, ich počet a veľkosť tabuľky sa nastaví automaticky, tak aby vyhovovali danej chybovosti a počtu položiek.

Z predošlého textu vypláva, že Bloomov filter má vopred danú pevne zvolenú veľkosť. Prúdové prostredie ale predpokladá potenciálne nekonečný počet prvkov. V tejto práci bol tento problém vyriešený zavedením dvoch Bloomových filtrov tzv. aktuálnym a záložným, ktoré sa medzi sebou striedajú. Princíp výmeny je zobrazený na obrázku 7.4, kde t_0 predstavuje počiatočný čas a V je požadovaná veľkosť Bloomovho filtra. Pre veľkosť filtra V budú vytvorené dva Bloomove filtre o veľkosti 2-krát väčšej teda $2V$, pričom veľkosť je udaná v sekundách, ktoré predstavujú časové okno reprezentujúce dočasnú históriu spojení. Proces výmeny prebieha nasledovne: Všetky toky patriace do intervalu $[t_0,\ t_0 + 2V]$ budú pridané do aktuálneho Bloomovho filtru (filter 1) a toky patriace do intervalu $[t_0 + V,\ t_0 + 2V]$ do záložného (filter 2). S príchodom toku s časovom väčším ako $t_0 + 2V$ dôjde k zmazaniu obsahu aktuálneho filtra (filter 1), ktorý je v tomto momente už plný a dôjde k jeho výmene s filtrom záložným (filter 2), ktorý je zaplnený len spolovice. Toky prichádzajúce v časovom intervale $(t_0 + 2V, \infty)$ budú vždy vložené do oboch filtrov a k výmene dôjde vždy po uplynutí V sekúnd. Je zrejmé, že prienik časových intervalov, ktoré oba Bloomove filtre zachytávajú má veľkosť V, čo zároveň predstavuje minimálne garantované časové okno zachytené históriou. Skutočná veľkosť Bloomovho filtra, respektíve očakávaný počet položiek je daný ako súčin predpokladanej intenzity príchodu tokov v tokoch za sekundu a dvojnásobku požadovanej veľkosti histórie v sekundách. Napríklad, ak je očakávaná intenzita

[5] http://www.partow.net/programming/bloomfilter/index.html
[6] https://github.com/CESNET/Nemea-Framework/blob/master/common/include/BloomFilter.hpp



Obr. 7.4: Proces výmeny Bloomových filtrov 1 a 2, obdĺžniky predstavujú zmenu funkcie uvedených filtrov v závislosti na čase.

spravovania 400 000 tokov za sekundu a história má zachytávať 120 sekúnd, potom bude veľkosť Bloomovho filtra daná hodnotou výrazu $400\,000 * 120 * 2$.

Do histórie spojení je vkladaná dvojica zdrojovej a cieľovej IP adresy každého prichádzajúceho toku s protokolom TCP, ICMP alebo UDP. Výnimku tvoria toky protokolu TCP s príznakmi RST alebo RST-ACK, ktoré do histórie vkladané nie sú. Dôvodom je princíp ustanovenia spojenia protokolu TCP, kedy si každá zo strán musí vymeniť tok s príznakom SYN. Preto ak nejaké zariadenie obdrží spätne rozptýlený tok s príznakmi ACK-SYN, potom odpoveď s príznakmi RST a RST-ACK nie je považovaná za obojsmernú komunikáciu, nakoľko dané príznaky indikujú, že táto komunikácia bola smerom k príjemcovi spätného rozptylu nevyžiadaná. Hoci sa táto práca zaoberá len spätným rozptylom v rámci protokolov TCP a ICMP sú do histórie pridávané aj UDP toky, pretože ICMP komunikácia mohla vzniknúť v súvislosti s komunikáciou protokolom UDP. Príkladom je ICMP správa s typom 3, ktorá môže vzniknúť pri neúspešnom pokuse o nadviazanie spojenia medzi stanicami používajúcimi protokol UDP.

Druhú dôležitú súčasť filtrovania obojsmernej komunikácie predstavuje dočasná pamäť, kde sú ukladané len toky s príznakmi typickými pre spätne rozptýlene pakety, zoznam týchto príznakov bol uvedený v kapitole 6. Dočasná pamäť je implementovaná ako prioritná fronta (`std::priority_queue`[7]) pričom najvyššiu prioritu majú najstaršie toky. Toky sú z dočasnej pamäte odstraňované po uplynutí aspoň V sekúnd od ich pridania. Hodnota V je identická s požadovanou veľkosťou histórie spojení.

Proces spracovania toku zachytáva výpis 7.1. Najskôr je spojenie práve obdržaného toku vložené do histórie spojení. História si udržiava aktuálny čas, ten je po vložení spojenia odovzdaný v návratovej hodnote. Na základe času histórie sú následne z dočasnej pamäte odobrané a spracované všetky toky staré aspoň V sekúnd. Pred spracovaním jednotlivých tokov dôjde ešte k overeniu existencie komunikácie v opačnom smere, a ak je tok jednosmerný bude predaný správcovi udalostí. V poslednom kroku je do dočasnej pamäte pridaný práve obdržaný tok a to v prípade, ak obsahuje príznaky typické pre spätný rozptyl.

Aktuálny čas, ktorým sa predošlí algoritmus riadi, je daný ako maximálny čas z tokov doposiaľ pridaných do histórie. V prezentovanej implementácii je za čas toku považovaný čas posledne prijatého paketu, ktorý daný tok tvorí. Okrem tohto času obsahuje tok aj čas

[7] https://en.cppreference.com/w/cpp/container/priority_queue

prvého paketu. V tejto práci je uprednostnený čas posledného paketu toku pred prvým z dôvodu menšieho rozdielu od času exportu toku zo sondy na kolektor. Čas posledného paketu v toku je taktiež bližšie skutočnému času v prípade, ak modul beží v reálnom čase.

```
1: if (!(nf.protocol == TCP && (nf.flags == RST || nf.flags == RSTACK))) {
2:     // Pridanie spojenia do historie spojeni
3:     h_time = history.add_connection(nf.src_ip, nf.dst_ip, nf.time_last);
4:     // Vyber tokov z docasnej pamate
5:     while (!buffer.empty() && (buffer.top().time_last + V <= h_time)) {
6:         flow = buffer.top();
7:         // Overenie obojsmernosti
8:         if(!history.contains(flow.dst_ip, flow.src_ip)) {
9:             event_tracker.add(flow);
10:        }
11:        buffer.pop();
12:    }
13: }
14: // Vlozenie tokov s priznakmi spatneho rozptylu do docasnej pamate
15: if (is_bs_like(nf) && buffer.size() < max_size) {
16:    buffer.push(nf);
17: }
```

Výpis 7.1 Pridanie spojenia práve obdržaného toku `nf` do histórie spojení `history` a postupný výber tokov `flow` z dočasnej pamäte `buffer`, za ktorým nasleduje vloženie obdržaného toku do dočasnej pamäte.

Veľkosť dočasnej pamäte by mala odpovedať súčinu intenzity príchodu tokov s príznakmi spätného rozptylu a požadovanou veľkosťou histórie, respektíve oneskorením V. Napríklad pri očakávanej intenzite 10 000 potenciálne rozptýlených tokov za sekundu bude veľkosť dočasnej pamäte daná ako $10\,000 * V$ tokov. V dátach poskytnutých od organizácie CESNET tvorili toky s príznakmi spätného rozptylu zhruba 2.3 % z celkového počtu tokov. Intenzitu potenciálne spätne rozptýlených tokov je teda možné určiť vynásobením celkovej intenzity príchodu tokov hodnotou $0,023$.

V prípade, ak skutočná intenzita spracovania tokov prevyšuje očakávanú intenzitu, dôjde k zvýšeniu počtu falošne pozitívnych prípadov v Bloomovej histórii. To znamená, že sa do časti spracovania dostane menej tokov s príznakmi spätného rozptylu, nakoľko budú nesprávne označené ako obojsmerné. Taktiež môže dôjsť k zahodeniu toku z dôvodu zaplienia dočasnej pamäte. Oba prípady spôsobia zanedbanie niektorých jednosmerných tokov, čo môže viesť k podhodnoteniu veľkosti DDoS útoku alebo k nemožnosti detekcie menších útokov, obojsmerné toky búdu v rámci daného časového okna V identifikované vždy správne.

Predpokladom použitia vyššie popísanej metódy určenia obojsmernosti spojenia je, aby vstupné dáta tokov obsahovali kompletnú komunikáciu zariadení v oboch smeroch. Uvedená metóda teda nie je vhodná pre zdroje dát, ktoré pokrývajú len jeden smer komunikácie, respektíve prítomnosť oboch smerov je závislá od iných faktorov, ako napríklad rôznosti smerovania paketov, kedy môže dôjsť k prenosu paketu mimo monitorovaný rozsah.

Sledovanie a export udalostí

Potenciálne spätne rozptýlené toky, ktoré sú na základe histórie spojení označené ako jednosmerné sú ďalej spracované do formy udalostí. Každá udalosť je vytvorená na základe skupiny tokov s rovnakým protokolom a zdrojovou IP adresou. Jednotlivé udalosti sú ucho-

vávané v tabuľke udalostí, ktorá je za účelom rýchleho prístupu implementovaná hašovacou tabuľkou (`std::unordered_map`[8]), kde kľúč predstavuje práve dvojica protokolu a zdrojovej IP adresy (adresa potenciálnej obete). Udalosť ďalej obsahuje zoznam rysov. Rysy predstavujú štatistické hodnoty, ktoré súhrne charakterizujú všetky toky tvoriace danú udalosť a sú navrhnuté tak, aby mali konštantnú pamäťovú zložitosť nezávislú na počte tokov, z ktorých sú tvorené.

Udalosť vzniká príchodom toku s unikátnou kombináciou protokolu a zdrojovej IP adresy, teda takej, ktorá v tabuľke udalostí ešte neexistuje. Udalosť môže byť exportovaná, respektíve zaslaná prostredníctvom knižnice TRAP na výstupné rozhranie v troch prípadoch:

1. Vypršaním pasívneho časovača – k udalosti nebol už dlhší čas pridaný žiadny ďalší tok. Indikuje koniec útoku.

2. Vypršaním aktívneho časovača – k udalosti sú stále pridávané nové toky, ale trvá už pomerne dlho a preto je priebežne exportovaná.

3. Okamžitý export – neobvyklý spôsob exportu, ktorý môže nastať len pri správnom ukončení modulu, teda obdržaním špeciálnej terminálnej správy z rozhrania TRAP. Okamžite dôjde k exportu všetkých záznamov z tabuľky udalostí.

Aktuálny čas, ktorým sa riadia časovače, je opäť daný ako maximálny čas zo skupiny tokov, ktoré vstúpili do fáze spracovania udalostí. V tomto prípade je potrebné podotknúť, že do tejto fázy vstupujú len toky z dočasnej pamäte. Dočasná pamäť je implementovaná ako prioritná fronta, kde prioritu určuje práve čas toku. Preto je v tejto fáze spracovania čas odvodený z maxima menej premenlivejší a určitým spôsobom presnejší ako v predošlých fázach nakoľko do tejto fázy vstupujú chronologicky zoradené toky. Uprednostnenie odvodenia času z tokov namiesto použitia reálneho času má dva dôvody. Odvodený čas umožňuje pracovať s tokmi aj v režime offline, kedy dáta neprichádzajú do modulu v reálnom čase. Druhý dôvod tvorí pridaná logika a zložitosť návrhu modulu. Pri riadení sa reálnym časom by bolo pravdepodobne nutné vytvoriť viacvláknový, resp. paralelný proces, ktorý by periodicky kontroloval platnosť časovačov v jednotlivých udalostiach.

Pasívny časovač je implementovaný formou prioritnej fronty (`std::priority_queue`[9]), ktorá je v rámci modulu označovaná ako kalendár. Pri vzniku udalosti je do kalendáru poznačený čas vzniku udalosti spolu s identifikovaným danej udalosti, pričom prioritu majú v kalendári položky s nižším časom. K overeniu pasívneho časovača dochádza pri zmene aktuálneho času v dôsledku spracovania toku. Pri overení časovača dôjde v prípade neprázdneho kalendára vždy aspoň k jednej kontrole, jedná sa o najstaršiu položku, a ak táto položka nesplní podmienku na vypršanie časovača, potom už nie je nutné kontrolovať ďalšie položky v kalendári, čo vyplýva z podstaty prioritnej fronty. V prípade vypršania časovača, ktorý sa vťahuje len k položke v kalendári, je potrebné overiť čas posledne pridaného toku do udalosti, ak je rozdiel medzi týmto časom a aktuálnym časom väčší ako hodnota pasívneho časovača, potom dôjde k exportu udalosti a jej zmazaniu z tabuľky udalostí, ak táto podmienka splnená nie je, dôjde len k naplánovaniu ďalšej kontroly do kalendáru. Útržok kódu reprezentujúci vyššie uvedenú funkcionalitu reprezentuje výpis 7.2, kde je počiatočná hodnota `stop_listing` nepravdivá, ak došlo k zmene aktuálneho času inak je pravdivá.

[8] https://en.cppreference.com/w/cpp/container/unordered_map
[9] https://en.cppreference.com/w/cpp/container/priority_queue

```
while (!stop_listing && !calendar.empty()) {
    calendar_item notification = calendar.top();
    event_iterator = events.find(notification.id);
    if (event_iterator == events.end()) {
        // Udalost bola exportovana aktivnym casovacom
        calendar.pop();
    } else {
        if (notification.time + passive_timeout < current_t) {
            calendar.pop();
            event_max_time = events[notification.id].get_max_time()
            if (event_max_time + passive_timeout < current_t) {
                // Exportovanie udalosti
                export(notification.id, PASSIVE);
                events.erase(notification.id);
            } else {
                // Naplanovanie novej kontroly
                calendar.push(calendar_item(notification.id, event_max_time));
            }
        } else {
            stop_listing = true;
        }
    }
}
```

Výpis 7.2 Implementácia pasívneho časovača

Aktívny časovač je implementačne jednoduchší a nevyžaduje žiadnu špeciálnu štruktúru. Ku kontrole dôjde vždy pred pridaním toku do udalosti ako zobrazuje výpis 7.3. V prípade aktívneho časovača však nedochádza k uvoľneniu položky z tabuľky udalostí ale len k opätovnej inicializácii rysov na predvolené hodnoty.

```
if (events[id].get_min_time() + active_timeout < current_t) {
    export(id, ACTIVE);
    events[id].clear();
}
```

Výpis 7.3 Implementácia aktívneho časovača

Výpočet rysov

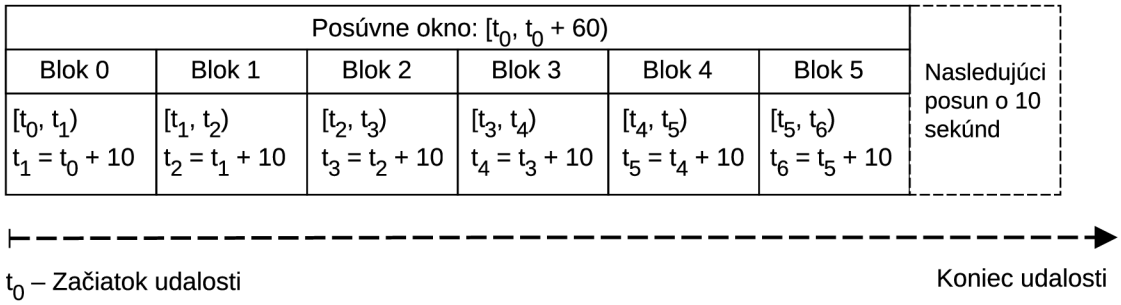
Táto sekcia popisuje proces výpočtu rysov uvedených v tabuľke 6.1 v kapitole 6. Krok výpočtu rysov prebieha vždy pri vložení toku do udalosti. Väčšina rysov má agregačný charakter, ako suma bajtov, či počet tokov udalosti. Pri vložení toku do udalosti potom stačí vykonať len príslušnú operáciu, v uvedenom príklade sumy operáciu sčítania. V prípade priemeru a štandardnej odchýlky je situácia podobná ako pri počte položiek či sume s tým rozdielom, že je typicky nutné vykonať ešte jednu finálnu úpravu. Napríklad priemer je možné jednoducho získať delením príslušnej sumy celkovým počtom prvkov v čase exportu. Štandardná odchýlka je vypočítaná principiálne rovnako ako priemer, nakoľko je možné previesť výraz, ktorý ju definuje na jednopriechodový ekvivalent a teda nie je potrebné uchovávať hodnoty jednotlivých tokov v pamäti. Uvedený prevod je zobrazený v rovnici 7.1. Z rovnice 7.1 vyplýva, že výpočet štandardnej odchýlky je možné previesť na výpočet odmocniny z rozdielu priemeru druhých mocnín hodnôt a druhej mocniny priemeru týchto hodnôt.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - E[X])^2} = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right) - E[X]^2} = \sqrt{E[X^2] - E[X]^2} \tag{7.1}$$

Problematické z hľadiska pamäťovej náročnosti sú rysy predstavujúce počet unikátnych položiek. Presný výpočet unikátnych položiek totiž vyžaduje reprezentovať v pamäti každú unikátnu hodnotu. Preto bol pre výpočet počtu unikátnych hodnôt použitý Bloomov filter. Toto použite sa principiálne zhoduje s históriou spojení, ktorá bola popísaná hneď v prvej časti tejto sekcie. Na výpočet unikátnych hodnôt sú teda použité dva Bloomove filtre, ktoré sa vzájomne striedajú. Na získanie počtu unikátnych hodnôt potom stačí jeden čítač, ktorý je inkrementovaný vždy keď sa daná hodnota v aktuálne používanom Bloomov filtri nenachádza a overovaná položka je v zápätí do filtru pridaná. Uvedený výpočet unikátnych hodnôt je heuristický a nemusí ponúkať presné výsledky, je však výhodný z hľadiska pamäťových nárokov. V rámci tejto práce bol Bloomov filter nastavený tak, aby zachytával zhruba polhodinovú históriu unikátnych hodnôt, nakoľko je väčšina DDoS útokov pomerne krátkych. Všetky udalosti zdieľajú ten istý aktuálny Bloomov filter, preto nie je overovaná unikátnosť len pre danú položku, ako napríklad čísla zdrojového portu, ale dochádza k overeniu unikátnosti n-tice, ktorá jednoznačne identifikuje udalosť, typ hodnoty a samotnú hodnotu, ktorej unikátnosť je predmetom overenia. Výpočet potrebnej veľkosti Bloomovho filtra je daný ako $V * I * R * 2$, kde V je požadovaná veľkosť histórie v sekundách, I je intenzita spracovania paketov s príznakmi typickými pre spätný rozptyl a R je počet rôznych rôznych rysov, ktoré používajú tento filter. Hodnota výrazu $V * I * R$ je potom ešte vynásobená dvomi, nakoľko $V * I * R$ reprezentuje veľkosť prekryvu Bloomových filtrov. Pri požadovaní veľkosti histórie zachytávajúcej 30 minút (1 800 sekúnd), intenzite 10 000 tokov za sekundu a pri použití 4 rysov, bude veľkosť histórie v počte položiek daná výrazom $1\,800 * 10\,000 * 4 * 2$. V tejto práci používajú uvedený princíp rysy unikátneho počtu cieľových portov, zdrojových portov, cieľovej IP adresy a cieľových podsietí s prefixom siete /24.

Posledným nepopísaným rysom z hľadiska spôsobu výpočtu je maximálny počet tokov za sekundu. Implementácia tohto rysu bola inšpirovaná implementáciou DDoS modulu `RS DoS` nástroja `corsaro`[10], ktorý je vyvíjaný organizáciou CAIDA[11]. Základom výpočtu je posuvné okno pokrývajúce 60 sekúnd diskretizované na šesť 10 sekundových blokov. Blok teda predstavuje určitý časový interval z posuvného okna. Vzťah bloku, posuvného okna a intervalu, ktorý pokrýva zobrazuje obrázok 7.5. Každý z blokov je možné reprezentovať ako osobitný čítač, posuvné okno teda predstavuje 6 čítačov. Algoritmus výpočtu potom funguje tak, že dôjde k inkrementovaniu hodnoty čítača odpovedajúcemu bloku, do ktorého časového intervalu patrí práve spracovávaný tok. Napríklad uvažujme situáciu z obrázku 7.5 s počiatočným časom udalosti t_0 a pozíciou okna $[t_0, t_0 + 60)$, potom ak má spracovávaný tok čas rovný $t_4 + 5$ bude inkrementovaný čítač bloku 4. Pri výbere bloku môžu nastať tri prípady. Prvý prípad predstavuje situáciu, keď je čas spracovávaného toku v intervale $[t_0, t_0 + 60)$, potom dôjde len k vyhľadaniu odpovedajúceho intervalu bloku a inkrementácii čítača. Ďalej sa môže stať, že čas toku bude menší ako čas začiatku okna, ktorý je v tomto prípade t_0, tieto toky sú mimo okna a nebudú pri spracovaní brané do úvahy, nemajú teda na okno a ani čítače žiadny efekt. Tento prípad však nie je pravdepodobný vzhľadom na to, že spracovávané toky pochádzajú z dočasnej pamäte a sú preto zoradené podľa času. Posledným prípadom je situácia, v ktorej je spracovávaný tok mimo okna, ale tentokrát

[10] https://www.caida.org/tools/measurement/corsaro/docs/corsaro__dos_8c.html
[11] https://www.caida.org/home/



Obr. 7.5: Časový interval posuvného okna, rozdelené časti okna predstavujú podintervaly, ktoré zachytávajú jednotlivé bloky.

z pravej strany, jeho čas je teda väčší, respektíve rovný $t_0 + 60$. V tom prípade dôjde k posunu okna o násobok veľkosti bloku, tak aby spracovávaný tok patril do posledného bloku okna. Pred posunom je však spočítaná suma všetkých čítačov jednotlivých blokov, ktorá reprezentuje počet tokov pre aktuálnu pozíciu posuvného okna. Výstupom algoritmu je maximálna hodnota počtu tokov v okne pre všetky rôzne pozície okna s krokom 10 sekúnd od počiatku až do konca udalosti. Posuvné okno bolo implementované kruhovým poľom o veľkosti 6, ktorého prvky reprezentujú čítače blokov.

Okrem rysov obsahuje udalosť aj informáciu o tom, na ktorý port obete bol vykonaný útok, ak sa jedná o protokol TCP, respektíve informáciu o použitej kombinácii ICMP typu a kódu. V rámci udalosti sú uchovávané 3 najpočetnejšie unikátne položky teda porty, respektíve kombinácie ICMP kódu a typu. Okrem samotných položiek je v udalosti uchovaná aj frekvencia ich výskytu. Problémom implementácie danej funkcionality je opäť potenciálne nekonečná množina unikátnych hodnôt položiek. Preto bol za týmto účelom použitý algoritmus *Frequent* [9]. Algoritmus *Frequent* bol navrhnutý na hľadanie frekventovaných položiek v prúdových dátach, ktorých výskyt presahuje $\frac{1}{k} * 100$ %. Pre dosiahnutie tohto cieľa je potrebných $k - 1$ čítačov. Chyba odhadu frekvencií jednotlivých položiek pri uvažovaní veľkosti vstupu n je maximálne ϵn, kde $\epsilon = \frac{1}{k}$. Respektíve, ak je maximálna prípustná chyba ϵ, potom je počet čítačov daný ako $k = \frac{1}{\epsilon}$.

Samotný algoritmus *Frequent* funguje nasledovne. Každý čítač ma pridelenú maximálne jednu položku a hodnota čítača vyjadruje frekvenciu asociovanej položky. Na začiatku algoritmu nie je pridelený žiadny čítač, nakoľko nebola spracovaná žiadna položka. Pri spracovaní položky dôjde k vyhľadaniu asociovaného čítača, ak existuje zvýši sa jeho hodnota o jedna, ak taký čítač neexistuje vyhľadá sa voľný čítač, ktorý sa nastaví na hodnotu jedna. V prípade, keď nie je so spracovávanou položkou asociovaný žiadny čítač a zároveň neexistuje ani voľný čítač dôjde k dekrementovaniu všetkých čítačov o hodnotu jedna. Po spracovaní všetkých položiek budú čítače s nenulovou hodnotou vyjadrovať frekvenciu asociovaných položiek.

V tejto práci bola zvolená prípustná chyba na úrovni 10 %, teda $\epsilon = 0{,}1$ a $k = 10$. Vzhľadom na malý počet čítačov je implementácia algoritmu *Frequent* realizovaná formou poľa dvojíc asociovanej položky a čítača v kombinácii so sekvenčným vyhľadávaním prvkov. Prípustná chyba sa môže zdať pomerne veľká, z princípu fungovania algoritmu ale vyplýva, že ak počet unikátnych položiek nebude vyšší ako 10, potom budú frekvencie presné. V praxi je navyše väčšina útokov smerovaných len na jeden port [11] a preto je vyššia miera teoretickej chyby prípustná.

Úvodné filtrovanie tokov

Ešte predtým ako sa toky dostanú do histórie a ďalšieho spracovania je hneď pri vstupe overený ich čas. Konkrétne sa jedná o čas posledného paketu, ktorý toku prináleží. Čas toku je totiž používaný na riadenia aktuálneho času v histórii spojení a v časti tvorby udalostí. Aktuálny čas je v uvedených prípadoch určení ako maximum z doposiaľ spracovaných tokov. Nakoľko sa jedná o maximum môže dôjsť k prudkým zmenám aktuálneho času smerom do budúcnosti, ktoré môžu byť spôsobné vznikom chyby na ceste od sondy k modulu. Prudké zmeny času nie sú z pohľadu funkcie modulu žiadúce a preto je možné nastaviť interval $[A-L, A+P]$, mimo ktorého budú všetky ostatné príchodzie toky ignorované. A predstavuje aktuálny čas modulu daný ako maximum z predchodzích tokov, vyjme práve spracovávaného toku, L a P sú nezáporné celé čísla, kde L definuje ľavé ohraničenie a P pravé ohraničenie uvedeného intervalu v sekundách. Na základe experimentov v kapitole 8 bola východzia hodnota L zvolená na 60 sekúnd a P na 30 sekúnd. Najdôležitejšie je obmedzenie hodnoty P, nakoľko toky v intervale $[A - L, A]$ nemajú vplyv na aktuálny čas v module.

V rámci procesu filtrovania položiek na základe času je nutné uvažovať prípadný výpadok zdroja prúdových dát. Pri výpadku dlhšom ako P sekúnd môže dôjsť k situácii, kedy bude modul zahadzovať všetky príchodzie toky, pretože nebudú patriť do intervalu $[A - L, A + P]$, čo znamená, že nedôjde k ich spracovaniu a ani aktualizovaniu aktuálneho času modulu. Z tohto dôvodu sleduje modul počet po sebe bezprostredne nasledujúcich odfiltrovaných tokov. V prípade, ak tento počet prekročí určitú hodnotu, dôjde k nastaveniu aktuálneho času na čas prvého toku, ktorý túto hodnotu prekročil. Následne taktiež dôjde k vymazaniu histórie spojení a dočasnej pamäte, nakoľko ich obsah nemusí byť vplyvom výpadku konzistentný vzhľadom na správne určenie vzájomnosti spojenia. Hodnota, po ktorej dôjde k uvedenej zmene je v module daná ako $V * I$, kde V je veľkosť histórie spojení v sekundách a I je očakávaná intenzita spracovania tokov v tokoch za sekundu.

7.3 Modul klasifikácie

Úlohou modulu klasifikácie je prijať udalosť z modulu extrakcie rysov a klasifikovať ju na základe jej rysov do kategórie DDoS pozitívnej/negatívnej kategórie. V prípade, že sa skutočne jedná o DDoS útok je udalosť nahlásená prostredníctvom platformy MISP[12] do projektu SPARTA[13]. Modul klasifikácie je implementovaný v jazyku Python3.

MISP

MISP je open-source platforma[14] pre ukladanie, zdieľanie a analýzu bezpečnostných hrozieb a indikátorov. Cieľom platformy je zjednotenie spôsobu nahlásenia a zdieľania kybernetických hrozieb medzi rôznymi bezpečnostnými skupinami, respektíve komunitami. Za účelom zjednotenia formy dát zavádza MISP preddefinované dátové typy a objekty. Medzi ďalšie funkcie platformy MISP patrí napríklad možnosť automatickej korelácie bezpečnostných udalostí a správa rozsahu zdieľania týchto dát medzi rôznymi komunitami.

[12] https://www.misp-project.org/
[13] https://sparta.eu/
[14] https://github.com/MISP/MISP

Realizácia modulu

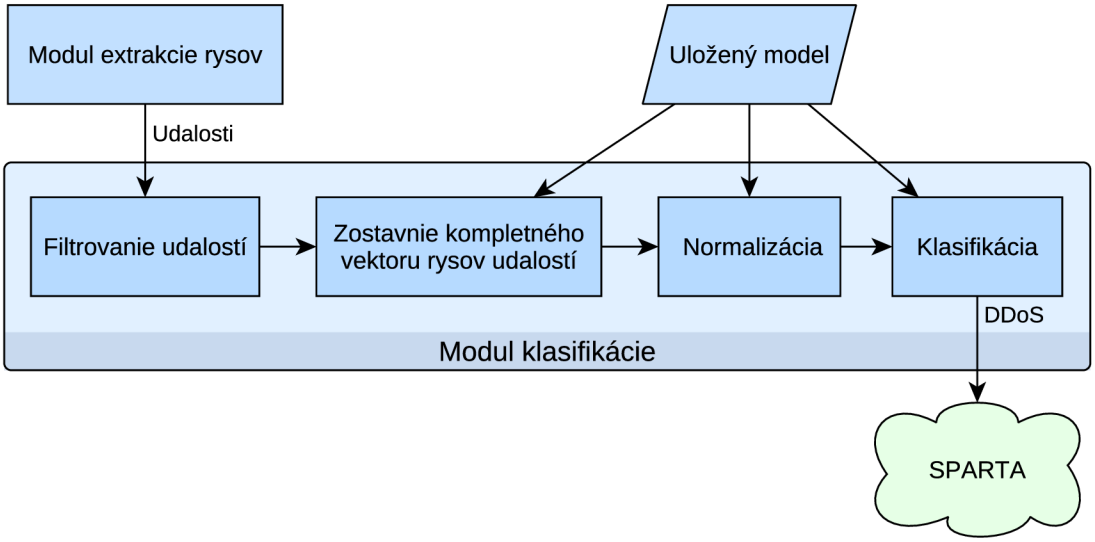
Vstupné udalosti sú do modulu klasifikácie zasielané prostredníctvom rozhrania TRAP. Po obdržaní udalostí dochádza k ich dodatočnému filtrovaniu. Filtrovanie má zamedziť nadbytočnému zasielaniu hlásení do projektu SPARTA nakoľko je počet DDoS útokov za deň v ráde tisícov. Filtračné kritérium je definované prostredníctvom parametrov modulu a je možné zvoliť obmedzenie vzhľadom na minimálny počet tokov útoku, jeho trvanie a rozhodovací prah použitý pri klasifikácii strojovým učením. Udalosti, ktoré vyhovujú daným filtračným kritériám potom prechádzajú do fázy predspracovania.

V prvom korku fázy predspracovania dochádza k výpočtu dodatočných rysov. Tieto rysy je možné získať z pôvodných rysov udalosti a preto nie sú spočítané v rámci modulu extrakcie rysov. Jedným z dodatočných rysov je napríklad priemerný počet tokov za sekundu daný ako pomer počtu tokov v udalosti a jej trvania. Na základe pôvodných a odvodených rysov je následne zostavený kompletný vektor rysov udalosti. Posledným krokom predspracovania je tvorba normalizovaného vektoru rysov, ktorý je následne predaný natrénovanému klasifikátoru. Ak je odpoveď klasifikátoru kladná, bude udalosť nahlásená do inštancie platformy MISP projektu SPARTA ako DDoS útok pomocou rozhrania `PyMISP`[15]. Kompletný zoznam informácií zaslaných prostredníctvom rozhrania MISP je nasledovný:

- IP adresa obete DDoS útoku
- Doménové meno IP adresy obete
- Číslo portu obete, v prípade útoku na viac portov je uvedený najfrekventovanejší
- Počiatok útoku – čas prvého paketu prvého toku udalosti
- Koniec útoku – čas posledného paketu posledného toku udalosti
- Trvanie útoku dané ako rozdiel medzi časom začiatku a konca útoku
- Odhadovaný počet bajtov, tokov a paketov útoku určený na základe rovnice 4.3
- Intenzita útoku v bitoch za sekundu určená z odhadovaného počtu bajtov
- Intenzita útoku v paketoch za sekundu určená z odhadovaného počtu paketov
- Poloha obete vyjadrená zemepisnou šírkou a dĺžkou
- Názov organizácie autonómneho systému IP adresy obete

Jednotlivé kroky modulu klasifikácie zobrazuje obrázok 7.6. Z obrázku je možné pozorovať závislosť medzi uloženým modelom a krokmi predspracovania. Túto závislosť zobrazuje daný diagram preto, lebo uložený model obsahuje okrem samotného modelu strojového učenia aj model normalizácie a zoznam použitých rysov. Zoznam použitých rysov je potrebný na správne zostavenie kompletného vektoru rysov a taktiež garantuje zachovanie poradia jednotlivých rysov vo vektore rysov, ktoré musí byť vzhľadom na natrénovaný model strojového učenia zhodné. Vyššie popísaný model je reprezentovaný prostredníctvom triedy jazyka Python3 a bol vytvorený v procese experimentovania ako znázorňuje obrázok 7.2. Na úschovu modelu z experimentov a jeho načítanie v module klasifikácie bola použitá knižnica `Pickle`[16]. Modul klasifikácie ďalej používa známu knižnicu strojového učenia `sklearn`[17],

[15] https://github.com/MISP/PyMISP
[16] https://docs.python.org/3/library/pickle.html



Obr. 7.6: Schéma modulu klasifikácie.

ktorá obsahuje potrebné definície modelu strojového učenia a modelu normalizácie. Pre reprezentáciu rysov vo forme vektoru vstupujúceho do strojového učenia je použitá knižnica `numpy`[18]. Na získanie informácie o polohe a organizácii obete útoku je použitá databáza služby MaxMind GeoIP[19], ku ktorej modul pristupuje prostredníctvom Python3 rozhrania `GeoIP2`[20].

[17] https://scikit-learn.org
[18] https://numpy.org/
[19] https://www.maxmind.com/en/geoip2-services-and-databases
[20] https://github.com/maxmind/GeoIP2-python

Kapitola 8

Vyhodnotenie implementovanej metódy

Táto kapitola sa venuje experimentálnemu vyhodnoteniu implementovanej metódy detekcie DDoS útokov zo spätného rozptylu navrhnutej v kapitole 6. V úvode tejto kapitoly sú popísané jednotlivé zdroje dát použité na uskutočnenie experimentov a vyhodnotenie metódy. Prvá séria experimentov používa dáta tokov a venuje sa hľadaniu vhodných parametrov modulu extrakcie rysov. Zvyšná časť experimentov pracuje s výstupom modulu extrakcie rysov – množinou udalostí, na ktorej je následne natrénovaný model strojového učenia. Záver kapitoly je venovaný zhodnoteniu výsledkov experimentov a možnostiam ďalšieho pokračovania.

Experimenty boli vykonané najmä v jazyku Python3 s použitím knižníc pre strojové učenie a prácu s dátami, ako `sklearn`[1], `imblearn`[2], `numpy`[3] či `pandas`[4] vo forme interaktívneho zošitu `Jupyter`[5]. Beh modelu extrakcie rysov bol uskutočnený s použitím výpočtových a úložných prostriedkov organizácie MetaCentum[6].

8.1 Zdroje dát

Ako bolo uvedené v kapitole 6, táto práca sa líši od existujúcich najmä tým, že pracuje mimo prostredia teleskopu a to navyše s dátami sieťových tokov. V dobe písania tejto práce sa nepodarilo nájsť takú dátovú sadu, ktorá by spĺňala obe požiadavky. Z tohto dôvodu bola v tejto práci vytvorená nová dátová sada s použitím reálnych dát poskytnutých od organizácie CESNET a CAIDA. Dáta organizácie CESNET predstavujú zdroj dát tokov a dáta organizácie CAIDA boli použité na anotovanie týchto dát, respektíve udalostí vytvorených z dát tokov.

[1] https://scikit-learn.org/
[2] https://imbalanced-learn.org/
[3] https://numpy.org/
[4] https://pandas.pydata.org/
[5] https://jupyter.org/
[6] https://metavo.metacentrum.cz

CAIDA

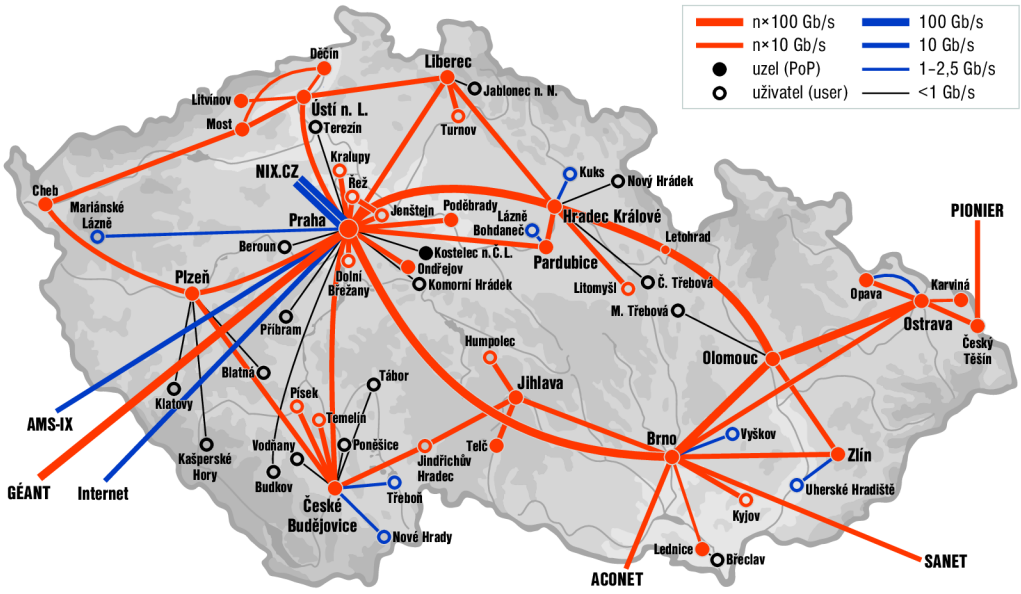
Organizácia CAIDA[7] má k dispozícii pomerne veľký sieťový teleskop s prefixom siete /8. Tento teleskop je vzhľadom na jeho veľkosť vhodným zdrojom dát pre analýzu spätného rozptylu a teda aj detekciu DDoS útokov [5, 16]. CAIDA poskytuje prístup k dátam tohto teleskopu v troch formách. Prvou možnosťou je použitie dát paketov, druhou dát tokov a tretiu možnosť predstavuje dátová sada DDoS útokov generovaná na dennej báze modulom RS DoS, ktorý bol popísaný v sekcii 5.1. V tejto práci boli za účelom anotácie použité dáta tokov, nakoľko modul RS DoS nerozlišuje typ použitého protokolu a dáta paketov sú pre potreby anotácie na základe príznakov nadbytočné. Dáta tokov teda poskytujú vhodnú mieru abstrakcie pre účely tejto práce. Tok je v podaní organizácie CAIDA konceptuálne podobný toku NetFlow a definuje ho osem [3] kľúčových položiek: zdrojová IP adresa, cieľová IP adresa, zdrojový port, cieľový port, protokol, TCP príznaky, TTL a veľkosť IP datagramu. K tejto n-tici potom tok obsahuje jednu nekľúčovú položku udávajúcu počet paketov tvoriacich tento tok. V dátovej sade tokov organizácie CAIDA [3] sú pakety spracovávané do tokov len v rámci pevne daných 60 sekundových intervalov a nie na základe pasívneho, či aktívneho časovača. Dva pakety teda tvoria jeden tok len v prípade, ak sa zhodujú ich kľúčové položky a zároveň patria do rovnakého časového intervalu. Pre anotáciu je potrebné získať len informáciu o tom, z ktorej IP adresy bol odosielaný spätný rozptyl (obeť útoku), preto boli dáta v rámci 60 sekundového intervalu ďalej agregované pomocou nástroja `cors-ft-aggregate`[8]. Agregovanie bolo uskutočnené na základe dvojice IP adresy obete a príznakov spätného rozptylu. Za agregačnú hodnotu bol zvolený počet rôznych cieľových IP adries, teda adries, na ktoré bol smerovaný spätný rozptyl v rámci teleskopu CAIDA. Táto hodnota bola uprednostnená pred počtom paketov, aby došlo k odstráneniu duplicitných paketov v podobe retransmisií. Tento prístup môže podhodnotiť celkový počet pozorovaných rozptýlených paketov na teleskope CAIDA. Toto podhodnotenie by však nemalo byť významné nakoľko je nepravdepodobné, aby bol spätný rozptyl zaslaný opakovane na tú istú adresu a to najmä pri menších útokoch.

CESNET

Organizácia CESNET[9] spravuje skoro milión IP adries, čo odpovedá jednej podsieti s prefixom siete /12. Meracie body, respektíve sondy, sú rozmiestnené na hranici tejto siete. Dáta tokov teda obsahujú len prevádzku, ktorá do tejto siete vstupuje alebo z nej vychádza, komunikácia v rámci siete nie je v dátach tokov prítomná. Topológiu siete CESNET zobrazuje obrázok 8.1. Na obrázku je možné pozorovať, že do siete CESNET vstupujú dáta z rôznych iných sietí, ako napríklad GÉANT[10] či SANET[11]. Monitorovanie pomocou tokov potom prebieha práve v mieste spojenia týchto sietí a siete CESNET.

V rámci tejto práce boli použité dve sady dát tokov. Prvá sada pokrýva 4 dni a bola zachytená v období od 6.8.2020 do 10.8.2020. Druhá sada pokrýva len 2 hodiny dát tokov zachytených 28.4.2021 vo frekventovanejšej časti dňa, konkrétne od 12:35 do 14:35. Prvá sada bude ďalej v texte a experimentoch označovaná ako primárna a druhá ako sekundárna. Dôvodom použitia dvoch sád je spôsob uchovania dát na kolektore. Dáta je možné zachytiť priamo z prúdu dát tokov vo formáte UniRec pomocou frameworku NEMEA alebo

[7] https://www.caida.org/home/
[8] https://www.caida.org/tools/measurement/corsaro/docs/tools.html#tool_corsagg
[9] https://www.cesnet.cz/
[10] https://www.geant.org/
[11] https://www.sanet.sk/



Obr. 8.1: Topológia siete CESNET [8].

je k dátam možné pristupiť skrz osobitné úložisko, kde sú dáta archivované. Prvý prístup používajúci formát UniRec má výhodu v tom, že prúd dát tokov obsahuje dáta zo všetkých meracích zariadení pokope. Tento spôsob je však vhodný len pre záchyt menšieho množstva dát tokov, nakoľko zariadenie, ktoré danú funkcionalitu poskytuje, nedisponuje dostatočným množstvom pamäte, pretože nie je určené na archiváciu dát tokov. Oproti tomu dokáže špeciálne úložisko uchovať až niekoľko mesiacov dát tokov. Dáta sú v ňom však organizované podľa jednotlivých meracích zariadení vo formáte nfcapd[12]. Pred prácou s týmito dátami je preto potrebné vykonať krok predspracovania, ktorý dáta z rôznych meracích bodov spojí dokopy. Spojením týchto dát sa však poruší časová charakteristika, respektíve poradie príchodu tokov na kolektor. Ako rozsah primárnej a sekundárnej dátovej sady naznačuje, primárna sada bola získaná zo špeciálneho úložiska a sekundárna zachytená prostredníctvom frameworku NEMEA. Primárna sada je v tejto práci použitá na účely trénovania modelu strojového učenia vzhľadom na to, že pokrýva väčší časový rozsah. Sekundárna sada, ktorá zachováva charakteristiku príchodu tokov, bola použitá na určenie vhodných parametrov modulu extrakcie rysov a to hlavne veľkosti histórie spojení a pasívneho časovača. Po určení týchto parametrov by už nemalo príliš záležať na poradí tokov v dátovej sade, nakoľko sú toky s príznakmi spätného rozptylu pred tvorbou udalostí v module extrakcie rysov zoradené.

Ako bolo uvedené vyššie, pri tvorbe primárnej sady bolo potrebné spojiť dáta rozdelené podľa meracích zariadení. Dáta sú v rámci zložky patriacej k meraciemu zariadeniu organizované do niekoľkých dielčích súborov, kde každý z nich obsahuje 5 minút dát tokov. K spojeniu týchto súborov potom došlo na základe prioritnej fronty. Prioritná fronta mala stanovenú pevnú veľkosť nastavenú tak, aby dokázala uchovať zhruba 5 až 10 minútový interval dát tokov zo všetkých meracích zariadení a po jej zaplnení dôjde k odobratiu najstaršieho toku a jeho uloženiu do výslednej dátovej sady tokov. Do tejto fronty bol po-

[12] https://github.com/phaag/nfdump

tom striedavým spôsobom vkladaný obsah dielčich súborov z rôznych meracích bodov, čím došlo k ich spojeniu pomocou zoradenia. V tomto procese bol za čas toku považovaný čas posledného paketu daného toku.

Z oboch sád tokov bola navyše v rámci experimentov odstránená tzv. tranzitná prevádzka. Jedná sa o toky popisujúce prevádzku, ktorá nepochádza zo siete CESNET a taktiež nie je jej cieľovou destináciou. Tranzitné toky sú teda také toky, kde ani jedno z koncových zariadení nepatrí do rozsahu IP adries, ktoré CESNET spravuje. Tieto toky sú nežiadúce, pretože pri nich nie je možné pozorovať kompletnú komunikáciu oboch zariadení nakoľko je prechod cez sieť CESNETu podmienený smerovaním. Bez tranzitnej prevádzky obsahuje primárna dátová sada takmer 38 miliárd tokov a sekundárna 1,1 miliardy.

8.2 Parametre modulu extrakcie rysov

V tejto časti sú popísané experimenty vedúce k zvoleniu vhodných parametrov modulu extrakcie rysov, ktorý bol popísaný v sekcii 7.2. Z hľadiska tvorby udalostí sú najdôležitejšími parametrami veľkosť histórie a pasívneho časovača. Nedostatočná veľkosť histórie môže viesť k nemožnosti správne identifikovať obojsmerné, respektíve jednosmerné toky a malá hodnota pasívneho časovača môže spôsobiť rozdelenie jednej udalosti na niekoľko menších udalostí. Oba parametre sú závislé na usporiadaní tokov a preto bola v tomto experimente použitá najmä sekundárna dátová sada zachytená priamo na kolektore, kde je poradie tokov zachované.

Voľba základných parametrov modulu

Medzi základné parametre modulu patrí: veľkosť prípustného časového okna vstupných tokov, očakávaný počet tokov za sekundu, prípustná miera falošne pozitívnych položiek v histórii spojení, očakávaný počet tokov s príznakmi spätného rozptylu a prípustná miera falošne pozitívnych položiek v histórii rysov.

Veľkosť prípustného časového okna určuje, ktoré toky budú ďalej spracované. Cieľom tohto parametru je odstránenie tokov s časom výrazne odlišujúcim sa od ostatných tokov. Ako bolo uvedené v sekcii 7.2, tento parameter pozostáva z dvoch hodnôt L a P definujúcich interval $[A-L, A+P]$, kde A je aktuálny čas modulu. Toky s časom mimo tento interval potom nie sú pri tvorbe udalostí brané do úvahy, pretože môžu negatívne ovplyvniť funkčnosť modulu. Hodnoty L a P boli v tejto práci určené na základe sekundárnej dátovej sady tokov a to pozorovaním rozdielu času prichádzajúceho toku a aktuálnej hodnoty času v module. Priemerná hodnota tohto rozdielu činila -32,7 sekundy so štandardnou odchýlkou 8,9 sekundy, minimálnou hodnotou rozdielu -334 a maximálnou 27 sekúnd. Výsledky indikujú, že vo väčšine prípadov je rozdiel v intervale $[-41,6; -23,8]$, teda čas väčšiny príchodzích tokov je zhruba o 30 sekúnd menší ako aktuálny čas v module. V tomto bode je potrebné podotknúť, že aktuálny čas v module je daný ako maximum z doposiaľ spracovaných tokov. Tieto výsledky sú pravdepodobne dôsledkom spôsobu exportu tokov na kolektor. Niektoré toky budú totiž exportované okamžite, napríklad na základe príznakov alebo zaplnením pamäte tokov na sondách, čo spôsobí aktualizovanie času modulu na čas blízky skutočnému času a následne sú exportované zvyšné toky oneskorené pasívnym časovačom. CESNET používa pasívny časovač s hodnotou 30-tich sekúnd, táto hodnota sa približne zhoduje s priemerným oneskorením nameraným v tomto experimente a je teda v súlade s hypotézou uvedenou vyššie. Vzhľadom na výsledky tohto experimentu bola hodnota P stanovená na 30 sekúnd, čo zároveň odpovedá maximálnemu možnému posunu aktuálneho času modulu. Pri voľbe

Parameter	Hodnota
Časové okno validných tokov	[A-60, A+30]
Očakávaný počet tokov v tokoch za sekundu	400 000
Očakávaný počet tokov s príznakmi spätného rozptylu v tokoch za sekundu	10 000
Očakávaná miera falošne pozitívnych položiek v histórii spojení	5 %
Očakávaná miera falošne pozitívnych položiek v histórii rysov	1 %

Tabuľka 8.1: Súhrn doporučených parametrov modulu extrakcie rysov. Premenná A predstavuje aktuálny čas v module.

ľavej strany intervalu L je potrebné uvažovať možné oneskorenie tokov v podobe pasívneho časovača. Hodnota L musí byť väčšia ako hodnota tohto časovača a preto bola stanovená na 60 sekúnd, čo je dva krát viac ako hodnota časovača použitého na CESNETe.

Očakávaný počet tokov za sekundu ovplyvňuje veľkosť Bloomovho filtra v histórii spojení a histórii rysov. V prípade, ak bude skutočná intenzita príchodu tokov väčšia ako očakávaná, dôjde k zvýšeniu miery falošne pozitívnych prípadov, teda dôjde k nadmernému filtrovaniu jednosmerných tokov, ktoré budú nesprávne označené ako obojsmerné. Pri meraní na sekundárnej sade bol priemerný počet tokov za sekundu 158 766 a z toho malo 2,2% príznaky spätného rozptylu, na primárnej sade bol priemerný počet nižší konkrétne 109 788 tokov za sekundu, z ktorých malo príznaky spätného rozptylu 2,3%. Rozdiel je zrejme spôsobený časom merania, nakoľko sekundárna sada pokrýva frekventovanú časť dňa a primárna sada obsahuje kompletné 4 dni. Na základe tohto merania bol v ďalších experimentoch nastavený parameter očakávanej intenzity tokov za sekundu na 400 000. Očakávaná intenzita tokov s príznakmi spätného rozptylu bola potom stanovená ako 2,5 % z hodnoty 400 000, teda na 10 000 tokov za sekundu. Uvedené hodnoty predstavujú zhruba dvojnásobok hodnôt nameraných na sekundárnej sade a to preto, aby nedošlo k nadmernému filtrovaniu tokov aj v prípade náhleho zvýšenia počtu tokov.

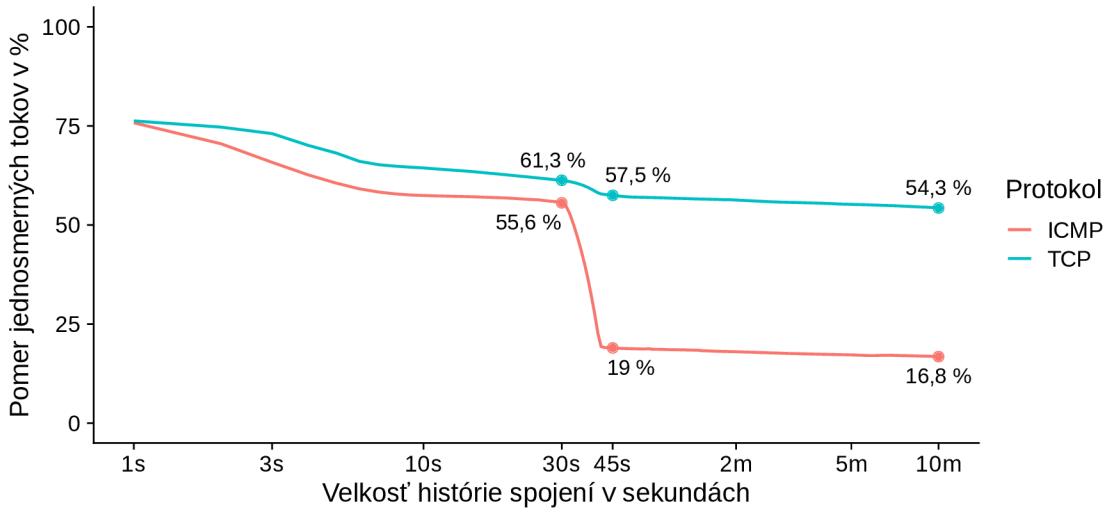
Miera prípustných falošné pozitívnych položiek bola v prípade histórie spojení stanovená na základe uváženia na hodnotu 5 % a pre históriu rysov na 1 %. Hoci sa môže zdať hodnota 5 % v prípade histórie spojení pomerne veľká, nemala by mať na funkciu modulu príliš veľký vplyv. Dôsledkom vyššej miery falošne pozitívnych položiek je len prípadné podhodnotenie veľkosti udalosti úmernej tejto hodnote. Čím je však miera prípustných falošne pozitívnych položiek nižšia, tým viac rastú výpočtové nároky, respektíve počet hašovacích funkcií v Bloomovom filtri, preto bola hodnota piatich percent zvolená ako kompromis medzi výpočtovou zložitosťou a presnosťou.

Tabuľa 8.1 sumarizuje doporučené základne parametre modulu extrakcie rysov v prostredí CESNET uvedených v rámci tohto experimentu.

Veľkosť histórie spojení

V tomto experimente bol vyhodnotený vplyv dĺžky histórie spojení na počet jednosmerných tokov s príznakmi spätného rozptylu. Prirodzene, čím dlhší časový úsek história spojení pokrýva, tým menší bude tento počet, pretože dochádza k vyhľadaniu opačného smeru spojenia

vo väčšom časovom úseku. Výsledky tohto experimentu sú zobrazené na obrázku 8.2. Miera

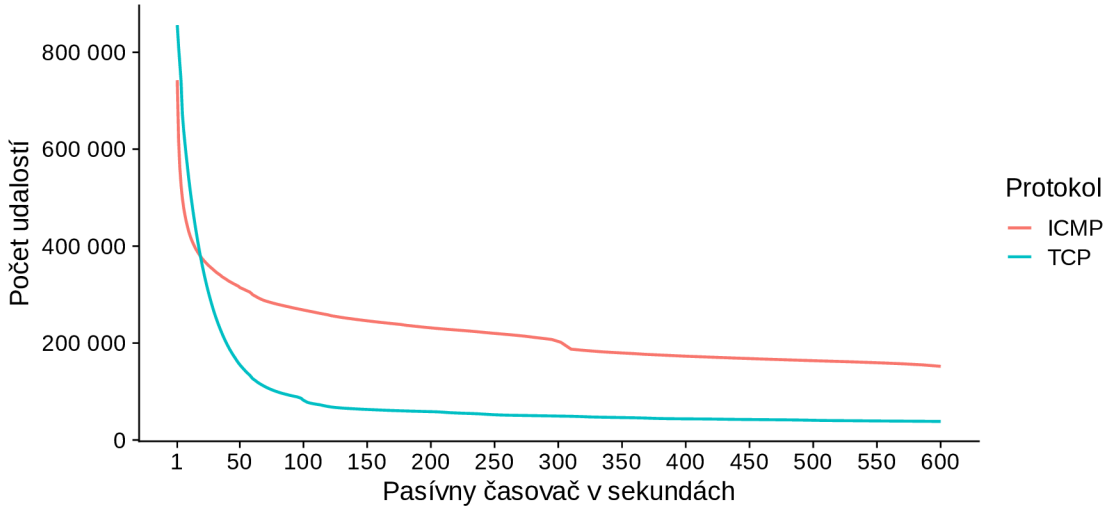


Obr. 8.2: Závislosť veľkosti histórie a pomeru jednosmerných tokov s príznakmi spätného rozptylu ku všetkým tokom s príznakmi spätného rozptylu.

jednosmerných tokov nie je vyjadrená v absolútnych číslach, ale pomerom jednosmerných tokov ku všetkým tokom s príznakmi spätného rozptylu, aby bolo možné porovnať priebeh na protokole TCP a ICMP. Na uvedenom grafe je možné pozorovať dva výrazne body. Prvým je bod s veľkosťou histórie jedna sekunda, ktorý oproti nepoužitiu žiadnej histórie (bod nula) redukuje počet jednosmerných tokov o 23,7 % na protokole TCP, respektíve o 24,2 % na protokole ICMP. Krivky oboch protokolov potom pozvoľne klesajú až ku hodnote 30-tich sekúnd, po ktorej dôjde k prudkej redukcii počtu jednosmerných tokov a to najmä na protokole ICMP, ďalej už nedochádza k výraznej zmene počtu jednosmerných tokov. Toto správanie pravdepodobne opäť odpovedá spôsobu exportovania tokov z NetFlow sond, kedy sú niektoré toky exportované okamžite a iné po uplynutí pasívneho časovača. Ako vhodná sa javí história s veľkosťou 45 a viac sekúnd. V tejto práci bola pre ďalšie experimenty použitá konzervatívna hodnota 120-tich sekúnd. Počas tohto experimentu dochádzalo len k zmene veľkosti histórie spojení a ostatné parametre modulu extrakcie rysov boli prevzaté z tabuľky 8.1.

Pasívny časovač

Podobne ako v prípade určenia histórie spojení bola vhodná hodnota pasívneho časovača získaná na základe experimentov so sekundárnou dátovou sadou. V tomto experimente bolo vyhodnotených niekoľko behov modulu extrakcie rysov s rôznou hodnotou pasívneho časovača, pričom veľkosť histórie spojení bola 120 sekúnd a ostatné parametre boli prevzaté z tabuľky 8.1. Výsledky jednotlivých behov zobrazujú obrázky 8.3 a 8.4. Obrázok 8.3 obsahuje vývoj počtu všetkých udalostí bez ohľadu na to koľko tokov ich tvorí. Oproti tomu graf na obrázku 8.4 zobrazuje len vývoj signifikantných udalostí pozostávajúcich z aspoň 30-tich tokov. V prvom prípade počet udalostí klesá exponenciálne a ustáli sa zhruba v okolí hodnoty časovača predstavujúcej 100 sekúnd. Na druhom obrázku je však vývoj dynamickejší a oproti prvému grafu počet udalostí aj rastie. Tento rast je možné vysvetliť tak, že pri zvýšení časovača dôjde k spojeniu menších udalostí, ktoré po spojení prekročia prah



Obr. 8.3: Vývoj počtu všetkých udalostí vzhľadom k veľkosti pasívneho časovača.

30-tich tokov. V určitom bode veľkosti časovača však počet udalostí už ďalej nemôže rásť nakoľko udalosti nie je možné spájať donekonečna. V tomto prípade dôjde k ustáleniu počtu udalostí až za hranicou 150-tich sekúnd a práve táto hranica sa javí ako vhodná hodnota pasívneho časovača. Ďalej je na uvedenom grafe možné pozorovať mierne stúpajúcu tendenciu v prípade vysokých hodnôt časovača na protokole ICMP. Tento trend indikuje, že zdroje týchto udalostí zasielajú pakety s pomerne nízkou intenzitou počas dlhého časového obdobia. To spôsobí, že aj pri pomerne vysokej hodnote časovača existujú ICMP udalosti, ktoré je možné spojiť čim dosiahnu hranicu 30-tich tokov.

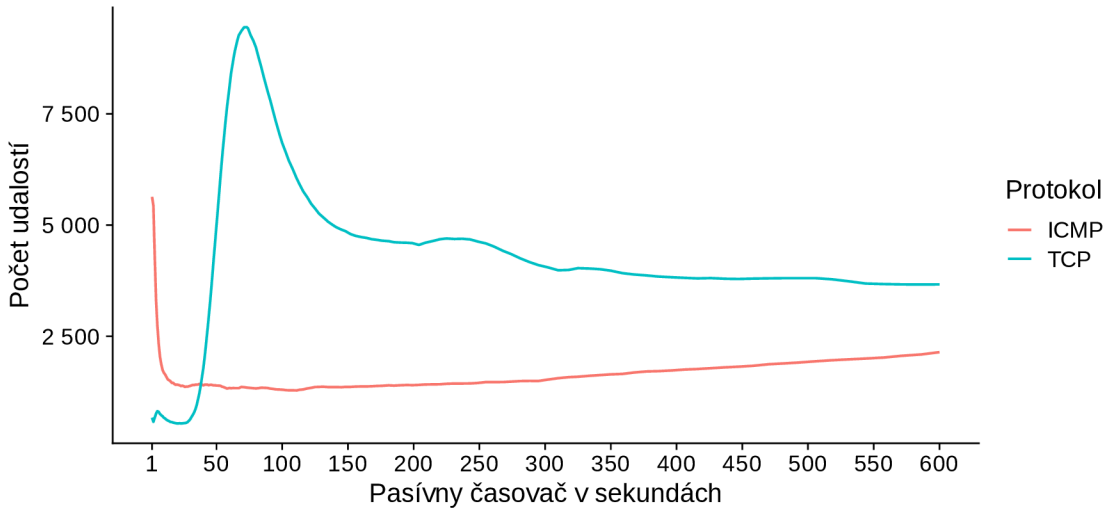
V tomto experimente nebol uvažovaný vplyv aktívneho časovača a jeho hodnota bola v rámci tohto experimentu nastavená tak, aby ním nemohla byť exportovaná žiadna udalosť. Počty udalostí teda zodpovedajú len nastaveniu pasívneho časovača. V tejto práci nebol overovaný vplyv aktívneho časovača na tvorbu udalostí pretože jeho hodnota záleží na požiadaviek frekvencie exportovania dlho trvajúcich udalostí.

V predošlých experimentoch nebola taktiež experimentálne stavená hodnota veľkosti histórie rysov, ktorá je použitá pre výpočet unikátnosti zdrojových adries a portov. Vzhľadom na to, že je väčšina útokov krátkych, bola táto hodnota nastavená podľa uváženia na hodnotu 30-tich minút.

8.3 Dátová sada udalostí

Dátová sada udalostí je tvorená množinou udalostí, respektíve rysov, ktoré boli získané modulom extrakcie rysov. Modul extrakcie bol spustený na primárnej dátovej sade tokov s parametrami odvodenými v predošlých experimentoch. Konkrétne sa jedná o parametre uvedené v tabuľke 8.1, ďalej bola veľkosť histórie spojení nastavená na 2 minúty, pasívny časovač na 2,5 minúty, aktívny časovač na 2 hodiny a veľkosť histórie rysov na 30 minút.

Nakoľko majú rysy udalostí štatistický charakter boli v dátovej sade ponechané len udalosti s aspoň 30-timi tokmi. Zvyšné udalosti boli následne anotované, pričom udalosť bola klasifikovaná ako DDoS útok vtedy, ak bolo možné súčasne nájsť v čase trvania udalosti taký minútový interval v dátovej sade tokov organizácie CAIDA, ktorý obsahoval minimálne 30 paketov s príznakmi spätného rozptylu zaslaných od rovnakého zdroja (po-



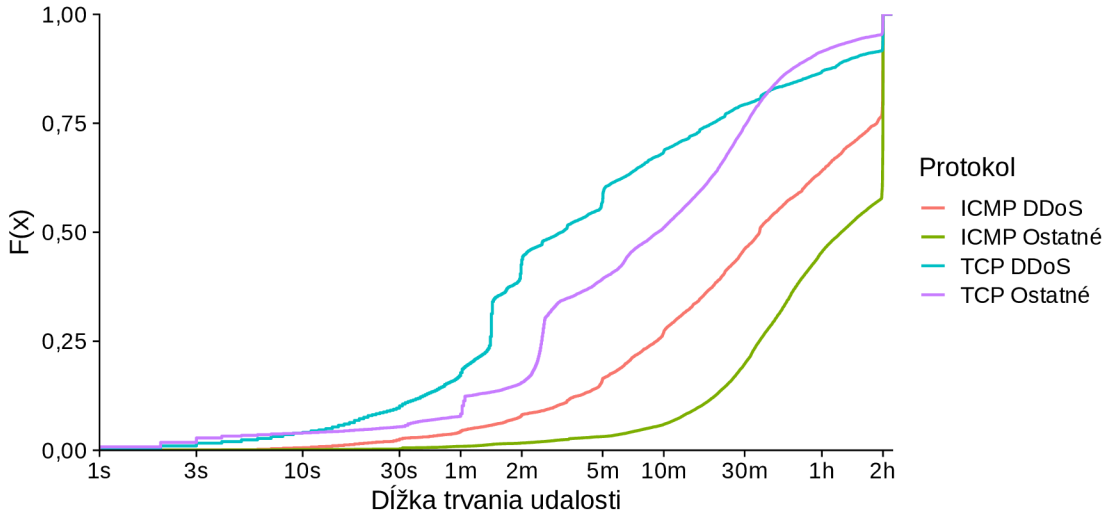
Obr. 8.4: Vzťah pasívneho časovača a počtu udalostí, ktoré pozostávajú z aspoň 30 tokov.

tenciálnej obete útoku). Súčasný výskyt nemusí byť presne synchronizovaný a pri anotácii je pripustená minútová tolerancia pred a po skončení útoku. Uvedené kritérium spĺňa 2 z 3 podmienok stanovených organizáciou CAIDA pre klasifikáciu DDoS útokov v module `RS DoS` [17]. V rámci tejto práce bol modul `RS DoS` popísaný v sekcii 5.1. Podmienky, ktoré autori stanovili znejú nasledovne:

1. Útok musí obsahovať viac ako 25 rozptýlených paketov
2. Útok musí trvať viac ako 60 sekúnd
3. Útok musel v nejakom bode dosiahnuť intenzitu väčšiu ako 30 paketov za minútu

Z uvedených podmienok nespĺňa kritérium použité v tejto práci podmienka 2, teda podmienku na minimálne trvanie útoku. Táto podmienka nie je splnená preto, lebo agregačná jednotka dátovej sady tokov organizácie CAIDA predstavuje jednu minútu, pakety teda mohli prísť v ľubovolnom podintervale v rámci tohto minútového intervalu. Trvanie útoku však bolo dodatočne obmedzené v rámci dát pozorovaných na CESNETe a to na hodnotu 30-tich sekúnd za účelom odstránenia ojedinelých položiek. Táto hodnota bola určená na základe grafu na obrázku 8.5, ktorý zobrazuje rozloženie dĺžky trvania udalostí, respektíve útokov vo forme distribučnej funkcie. Na uvedenom grafe je možné pozorovať pomerne málo udalostí s trvaním v rozsahu medzi jednou až tridsiatimi sekundami a potom dochádza k pozvoľnému rastu daných kriviek. Z kriviek je ďalej možné pozorovať, že najkratšie udalosti obsahuje trieda *DDoS* útokov protokolu TCP a až 40,6 % z týchto útokov má dĺžku medzi jednou a až piatimi minútami.

Kritérium zavedené v tejto práci splnilo v rámci dátovej sady CAIDA 4 192 unikátnych IP adries obetí útoku a z toho bolo na CESNETe pozorovaných 2 629 (63 %) v rámci protokolu TCP. V prípade protokolu ICMP bolo na teleskope CAIDA pozorovaných 2 576 unikátnych IP adries obete a z nich 1 191 (45 %) na CESNETe. Rozdiel v počte pozorovaných útokov je zrejme spôsobený väčším rozsahom teleskopu CAIDA, ktorý je zhruba 18 krát väčší ako CESNET a teda umožňuje pozorovať menšie DDoS útoky. Ďalej boli pri výpočte unikátnych IP adries na CESNETe brané do úvahy len udalosti v dátovej sade,



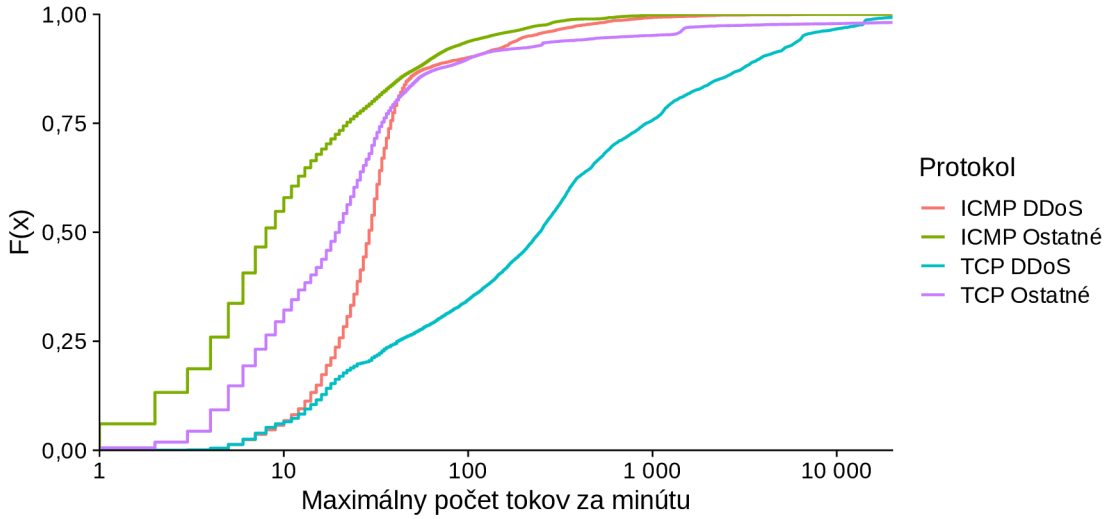
Obr. 8.5: Distribučná funkcia dĺžky trvania udalostí pozorovaných v rozsahu organizácie CESNET. Strmý nárast v bode dvoch hodín odpovedá veľkosti aktívneho časovača a teda aj maximálnej možnej zaznamenaj veľkosti útoku.

teda tie, ktoré majú aspoň 30 paketov a trvajú aspoň 30 sekúnd. Tieto filtračné kroky mohli preto taktiež podhodnotiť celkový počet vzájomne pozorovaných IP adries obetí.

V dátovej sade udalostí bol v rámci TCP útokov najčastejšie cieľom útoku port 443 a to v 32,8 % udalostiach, za ním nasleduje port 80 (19,5 %) a 22 (5,1 %). Pri protokole ICMP sa často vyskytovali nasledujúce dvojice typu a kódu: 3.3 (79,1 %), 11.0 (6,6 %) a 0.0 (5,8 %), pričom dvojica 3.3 (nedostupný port) indikuje, že sa pravdepodobne jednalo o UDP útok, ktorý vyvolal danú ICMP komunikáciu [1].

Obrázok 8.6 ďalej zobrazuje najväčšiu dosiahnutú intenzitu zasielania tokov na rozsah organizácie CESNET v podobe metriky maximálneho počtu tokov za minútu. Najmenšiu intenzitu majú udalosti v triede *ostatné* ako pri protokole TCP tak aj ICMP a najvyššiu trieda *DDoS* útokov protokolu TCP.

Okrem vyššie prezentovanej dátovej sady udalostí bola vytvorená ešte jedna dátová sada udalostí, ktorej postup tvorby a anotácie sa zhoduje s prvou sadou až na jednu výnimku. Modul extrakcie rysov bol pri tvorbe druhej dátovej sady udalostí upravený tak, aby periodicky po spracovaní 2 miliárd tokov, čo je zhruba každých 5 hodín, prestal správne pracovať na 2 až 10 minút. Počas tohto časového intervalu nedochádzalo k filtrovaniu obojsmerných tokov, čím sa do fázy tvorby udalostí dostali aj toky, ku ktorým existuje opačný smer. Úlohou druhej dátovej sady je simulovanie výpadkov dát tokov, respektíve meracieho zariadenia, ktoré môžu spôsobiť nesprávne určenie obojsmernosti spojenia. Takto vytvorená sada bola potom použitá na trénovacie účely a dátová sada bez prerušení bola použitá na testovanie a validáciu modelu. Bližší popis rozdelenia dátových sád na základe účelu rozoberá sekcia 8.4. Tabuľka 8.2 uvádza počty udalostí pre jednotlivé triedy a dátové sady. Z tabuľky je možné pozorovať, že zavedením prerušení sa zvýšil hlavne počet položiek v kategórii *ostatné* a v kategórii DDoS útokov zostal počet udalostí takmer nezmenení. Z pohľadu unikátnych IP adries obsahovala druhá dátová sada v triede *DDoS* len jednú novú obeť a zvyšné IP adresy obetí boli identické s prvou sadou. Toto porovnanie sád zároveň potvrdzuje predpoklad jednosmernej povahy spätného rozptylu, nakoľko sa položky v triede *DDoS* takmer vôbec nezmenili.



Obr. 8.6: Distribučná funkcia maximálneho počtu tokov zo minútu pozorovaného v rámci rozsahu organizácie CESNET.

Protokol	DDoS		Ostatné	
	Prvá sada	Druhá sada	Prvá sada	Druhá sada
TCP	7 217	7 218	15 454	51 133
ICMP	7 004	7 004	44 503	87 109

Tabuľka 8.2: Počet udalostí v rámci daných tried. Prvá sada udalostí predstavuje sadu bez zmeny funkcie modulu extrakcie rysov, narozdiel od druhej sady, ktorá obsahuje niekoľko simulovaných výpadkov meracieho bodu.

8.4 Trénovanie a vyhodnotenie klasifikácie DDoS útokov zo spätného rozptylu

Táto sekcia popisuje proces trénovania a vyhodenia klasifikácie DDoS útokov s využitím strojového učenia. Nakoľko sú dátové sady nevyvážené, boli pre vyhodnotenie modelu použité metriky F1 skóre, PPV (*precision*) a citlivosť (*recall*), ktoré sú vysvetlené v rámci kapitoly 5. Výstupom trénovania sú dva modely strojového učenia, jeden pre protokol TCP a druhý pre protokol ICMP.

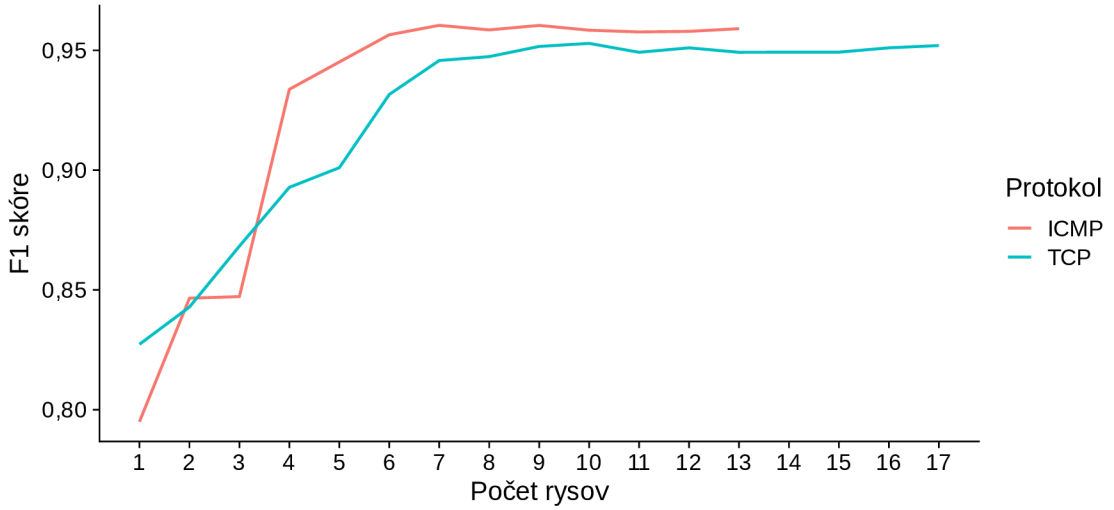
V prvom kroku bola rozdelená prvá a druhá dátová sada udalostí do častí podľa ich účelu v procese trénovania a vyhodnotenia. Výsledné rozdelenie zobrazuje tabuľka 8.3. Z tabuľky je možné pozorovať, že prvé dva dni zo zachytenej prevádzky boli vyhradené na trénovacie účely, nasledujúci deň na validačné a posledný deň na finálne vyhodnotenie modelu. Trénovacia sada bola použitá len na natrénovanie modelu strojového učenia a validačná na vyhodenie natrénovaného modelu voči rôznym nastaviteľným parametrom trénovaného modelu (hyperparametre). Testovacia sada je potom použitá až vo fáze vyhodnotenia finálneho modelu vybraného na základe výsledkov dosiahnutých na validačnej sade. Vyhodnotenie modelu prebiehalo v rámci prvej dátovej sady udalostí nakoľko odpovedá behu modulu extrakcie rysov bez zámerného narušenia jeho funkcionality. Za účelom zvýšenia robustnosti modelu bola však na tvorbu trénovacej sady použitá druhá sada udalostí, ktorá obsahuje simulované výpadky filtrovania obojsmerných tokov.

Účel sady	Zdroj	Časové obdobie	TCP		ICMP	
			DDoS	Ostatné	DDoS	Ostatné
Trénovanie	Druhá sada udalostí	08.06 – 08.07.2020	3 976	28 282	4 045	48 225
Validácia	Prvá sada udalostí	08.08.2020	1 749	2 678	1 322	8 154
Testovanie	Prvá sada udalostí	08.09.2020	1 432	3 239	1 507	8 683

Tabuľka 8.3: Rozdelenie prvej a druhej dátovej sady udalostí na trénovaciu, validačnú a testovaciu sadu.

Po rozdelení boli jednotlivé dátové sady normalizované pomocou triedy `StandardScaler`[13]. Motiváciou použitia normalizácie je prevedenie rysov do štandardizovaného tvaru vhodného pre algoritmy strojového učenia. Uvedený typ normalizácie funguje tak, že odčíta od každého rysu jeho priemernú hodnotu a následne túto hodnotu predelí štandardnou odchýlkou.

V ďalšom kroku bol vykonaný výber signifikantných rysov a to na základe metódy rekurzívnej eliminácie (`RFE`[14]). Princíp metódy rekurzívnej eliminácie spočíva v iteratívnom odstraňovaní najmenej významných rysov s použitím dodaného modelu strojového učenia. Odstraňovanie jednotlivých rysov prebieha až dovtedy, kým nie je dosiahnutá predom stanovená hranica žiadaného počtu rysov. V tejto práci bol ako interný model `RFE` použitý model náhodných lesov, ktorý poskytuje atribút určujúci dôležitosť jednotlivých rysov a práve týmto atribútom sa riadi proces výberu rysov v procese rekurzívnej eliminácie. Graf



Obr. 8.7: Hodnota F1 skóre na validačnej dátovej sade vzhľadom na počet rysov vybraných metódou rekurzívnej eliminácie.

8.7 zobrazuje vzťah medzi počtom rysov vybraných metódou `RFE` voči metrike F1 skóre vyhodnotenej na validačnej dátovej sade. Z uvedeného grafu je možné pozorovať, že k ustáleniu hodnoty F1 skóre dôjde pri deviatich rysoch v prípade protokolu TCP a pri siedmich

[13] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[14] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

Klasifikátor	TCP			ICMP		
	F1 skóre	PPV	Citlivosť	F1 skóre	PPV	Citlivosť
Gradient Boosting	97,3	96,2	98,3	96,4	94,2	98,7
Náhodne lesy	96,3	94,8	97,8	94,6	91,4	98,0
K najbližších susedov (KNN)	95,4	93,4	97,5	94,7	92,1	97,5
Rozhodovací strom	95,1	96,7	93,5	94,5	91,6	97,6
SVC	94,0	91,1	97,1	94,2	90,9	97,9

Tabuľka 8.4: Vyhodnotenie rôznych druhov klasifikátorov v rámci validačnej dátovej sady prostredníctvom metriky F1 skóre, prediktívnej hodnoty pozitívneho testu (PPV) a citlivosti v %.

na protokole ICMP. Na základe tohto experimentu bolo trénovanie v ďalších fázach uskutočnené na 12-tich rysoch v rámci protokolu TCP a 9-tich pri protokole ICMP. Tieto počty rysov sú o niečo väčšie ako hodnoty ustálenia z obrázku 8.7 a to preto, lebo uvedené množiny rysov boli získané len z predbežného modelu. Tento výber rysov bol teda urobený za predpokladu, že finálny model potenciálne využije viac rysov ako predbežný. Kompletný zoznam rysov bol uvedený v kapitole 6 v tabuľke 6.1. Vybrané množiny rysov sú uvedené v prílohe A v tabuľke A.1 pre protokol TCP a tabuľke A.2 pre protokol ICMP. Okrem vymenovania jednotlivých rysov obsahujú dané tabuľky taktiež dôležitosť týchto rysov vzhľadom k finálnemu modelu strojového učenia získanú prostredníctvom atribútu `feature_importances_`. Ako najpodstatnejšie z hľadiska modelu sa ukázali rysy spojené s unikátnosťou položiek a to najmä v podobe počtu unikátnych cieľových IP adries a podsietí s prefixom /24.

Trénovacia dátová sada bola ďalej vyvážená kombináciou generovania nových udalostí triedy *DDoS* metódou `SMOTE`[15] a náhodným vzorkovaním triedy *ostatné*, tak aby bolo v každej triede 20 000 položiek pre protokol TCP ako aj ICMP. V tomto bode je potrebné podotknúť, že validačná spolu s testovacou dátovou sadou vyvážené neboli a to preto, aby odrážali skutočné rozloženie tried udalostí v reálnej prevádzke. Trénovacia sada bola vyvážená z dôvodu porovnania rôznych modelov strojového učenia, ktorým by nevyvážaná sada nemusela vo fáze trénovania vyhovovať. Po vyvážení trénovacej sady došlo k vyhodnoteniu niekoľkých typov modelov strojového učenia, pričom pre každý typ modelu bolo vyskúšaných niekoľko parametrov s použitím mriežkového vyhľadávania (`GridSearchCV`[16]). Výsledky najlepších modelov v rámci daného typu obsahuje tabuľka 8.4, pričom kritériom posúdenia kvalita bola hodnota F1 skóre. Z pohľadu tejto metriky dopadol najlepšie klasifikátor Gradient Boosting a to v prípade TCP ako aj ICMP. Pre tento klasifikátor boli preto ďalej hľadané vhodné hyperparametre s použitím trénovacej a validačnej sady opäť v kombinácii s mriežkovým vyhľadávaním. Tentokrát však bola v mriežkovom vyhľadávaní optimalizovaná hodnota PPV, ktorá je tým lepšia čím menšia je chyba typu falošne pozitívny. Falošne pozitívne chyby sú nežiadúce z pohľadu nahlásenia udalostí do externých systémov, nakoľko dôjde k nahláseniu udalosti ako DDoS útoku aj keď ním v skutočnosti nie je, preto sú prípustnejšie chyby typu falošne negatívny, kedy útok nebude detegovaný. Algoritmus Gradient Boosting poskytuje okrem samotnej binárnej klasifikácie taktiež možnosť získať skóre určujúce istotu klasifikácie. Hodnota tohto skóre je v rozmedzí od nula do jedna, kde

[15] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
[16] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Dátová sada	Rozhodovací prah	TCP			ICMP		
		F1 skóre	PPV	Citlivosť	F1 skóre	PPV	Citlivosť
Testovacia	0,50	95,8	95,4	96,1	96,2	94,6	97,9
	0,70	95,8	96,7	95,0	96,1	95,2	96,9
	0,90	95,2	97,7	92,9	95,5	95,6	95,4
	0,95	94,5	98,0	91,2	95,4	96,1	94,7
	0,99	91,2	98,4	85,0	94,1	97,0	91,4
Validačná	0,50	97,3	96,7	97,9	96,6	94,4	98,8
	0,70	97,2	97,3	97,1	96,6	95,1	98,2
	0,90	96,7	98,2	95,3	96,5	95,8	97,1
	0,95	96,0	98,3	93,9	96,2	96,5	95,8
	0,99	93,5	98,7	88,9	94,6	97,6	91,8

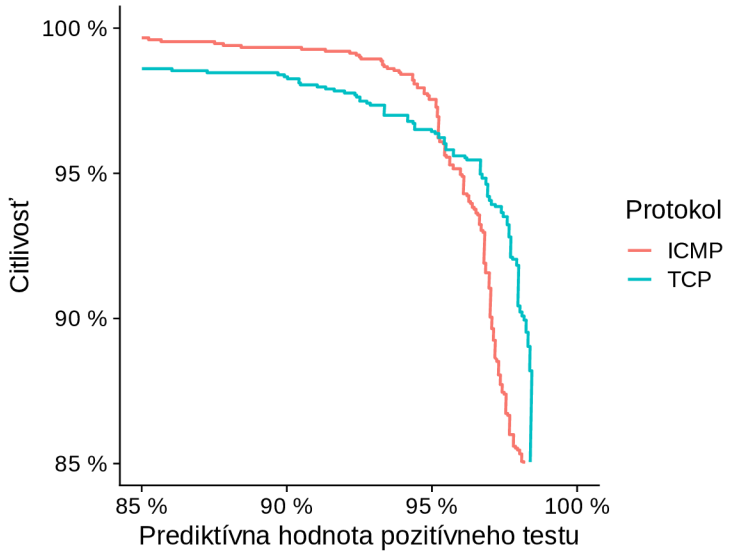
Tabuľka 8.5: Vyhodnotenie finálneho klasifikátoru na testovacej a validačnej sade vzhľadom na rôzne hodnoty rozhodovacieho prahu voči metrikám F1 skóre, prediktívnej hodnote pozitívneho testu (PPV) a citlivosti v %.

vyššie hodnoty tohto skóre predstavujú vyššiu istotu správnosti klasifikácie, ak je navyše rozloženie tried udalostí v trénovacej sade identické reálnemu použitiu, potom toto skóre vyjadruje pravdepodobnosť správnosti zaradenia udalosti do danej triedy. S použitím tohto skóre je teda možné zvoliť rozdeľovací prah, kde po prekročení tohto prahu budú všetky udalosti zaradené do kategórie *DDoS* a zvyšné do kategórie *ostatné*. Rozdeľovacím prahom je preto možné ovplyvňovať hodnotu PPV a citlivosti. Tento vzťah zobrazuje tabuľka 8.5, kde je vyhodnotený finálny model vzhľadom k rôznym hodnotám rozhodovacieho prahu. Prirodzene, čím vyššia je hodnota prahu, tým lepšia je hodnota metriky PPV a menšia citlivosť. Vzťah citlivosti a PPV pre rôzne hodnoty prahu zobrazuje taktiež obrázok 8.8, kde je možné vizuálne pozorovať kompromis (*tradeoff*) medzi hodnotou citlivosti a PPV.

8.5 Zhrnutie výsledkov

V tejto kapitole boli experimentálne stanovené vhodné parametre modulu extrakcie rysov. Medzi najdôležitejšie parametre tohto modulu je možné zaradiť veľkosť histórie spojení a veľkosť pasívneho časovača. Experimenty ukázali, že časový úsek, ktorý má história spojení pokrývať, musí byť väčší ako veľkosť pasívneho časovača na meracích zariadeniach, aby došlo k optimálnemu odfiltrovaniu obojsmerných tokov. V prípade pasívneho časovača sa javí ako vhodné požitie hodnoty dvoch a pol minút, kedy sa počet udalostí ustáli.

Modul extrakcie rysov bol následne s vhodnými parametrami spustení na 4 dňoch dát tokov z reálnej prevádzky organizácie CESNET. Výstup tohto modulu v podobe množiny udalostí bol následne anotovaný s použitím dát organizácie CAIDA do kategórie *DDoS* útokov a *ostatné*. Anotované dáta boli v ďalšom kroku použité na vytvorenie modelu detekcie DDoS útokov, pričom prvé dva dni dát boli použité na trénovanie, nasledujúci deň na validáciu a posledný deň na testovanie. Spomedzi modelov sa ako vhodné javí použitie algoritmu Gradient Boosting, ktorý dosiahol na testovacej sade s predvoleným rozhodovacím prahom 0,5 hodnotu F1 skóre 95,8 %, PPV 95,4 % a citlivosti 95,4 % na protokole TCP, respektíve F1 skóre 96,2 %, PPV 94,6 % a citlivosť 97,9% na protokole ICMP.



Obr. 8.8: Vzťah citlivosti a prediktívnej hodnoty pozitívneho testu (PPV) pre rôzne hodnoty rozhodovacieho prahu finálnych modelov strojového učenia protokolu TCP a ICMP. Hodnoty boli získané vyhodnotením na testovacej dátovej sade.

Hoci sú výsledky modelu pomerne uspokojivé, stále ponúkajú priestor na zlepšenie. Najjednoduchším riešením s cieľom zlepšenia modelu je rozšírenie dátovej sady udalostí na väčší časov úsek v ráde týždňov či mesiacov. Toto rozšírenie by nemalo byť príliš pamäťovo náročné, pretože vyžaduje len spustenie modulu extrakcie rysov na prúdových dátach tokov a uloženie extrahovaných udalostí. Vo fáze implementovania tohto modulu bolo však potrebné mať k dispozícii kompletné dáta tokov pre zvolený interval dátovej sady udalostí. Dôvodom bolo umožnenie opakovaného spustenia modulu v prípade jeho úprav, nakoľko je modul už vo finálnej podobe je možné vytvoriť väčšiu dátovú sadu v jednom priechode. Zložitejším spôsobom zlepšenia modelu je zavedenie nových rysov. Nájsť nové rysy nie je jednoduché vzhľadom na obmedzené množstvo informácií, ktoré obsahujú toky. Nové rysy si preto musia vystačiť len s kombináciou informácií z viacerých tokov vo forme priemerov či odchýlok. Inou možnosťou získania nových rysov môže byť aktívne dotazovanie služby počas potenciálneho priebehu útoku s cieľom nadobudnutia dodatočných informácií o tejto službe. Z pohľadu ďalšej analýzy a anotovania môže byť zaujímavá práca s paketovými dátami, ktoré teleskop organizácie CAIDA taktiež poskytuje. Tieto dáta by mohli byť napríklad použité na anotovanie spätného rozptylu protokolu UDP, ktorý bol v tejto práci zachytený len v rámci protokolu ICMP.

Kapitola 9

Záver

Sieťové teleskopy sú vzhľadom na absenciu legitímnej komunikácie vhodným prostriedkom na sledovanie a analýzu nelegálnej aktivity v podobe scanov a spätného rozptylu. Teleskopy však majú značnú nevýhodu, ktorá vyplýva z ich definície, jedná sa o veľké nepriradené adresové priestory, kde nie sú prítomne žiadne zariadenia. Dnes je adresový priestor IPv4 pomerne vzácny a vytvorenie vlastného teleskopu je z tohto dôvodu problematické. Táto práca sa preto venovala možnostiam detekcie DDoS útokov s využitím spätného rozptylu v bežných sieťach, kde sú prítomné legitímne zariadenia a to navyše len s použitím abstrahovaných dát sieťovej komunikácie v podobe tokov.

V tejto práci bola postupne vysvetlená problematika sieťových tokov, DDoS útokov a spätného rozptylu. Z problematiky sieťových tokov bola predstavená architektúra NetFlow s dôrazom na spôsob prevodu paketových dát do podoby tokov. Ďalej boli v tejto časti práce vysvetlené výhody a nevýhody reprezentácie dát v podobe sieťových tokov z pohľadu ich použitia vzhľadom k následnej analýze, detekcii hrozieb a monitorovaniu siete. Práca sa následne venovala DDoS útokom a uviedla niekoľko najznámejších typov týchto útokov, medzi ktoré patrí napríklad zahltenie SYN paketmi na protokole TCP. Dôležitú súčasť problematiky DDoS útokov predstavuje mechanizmus podvrhnutia IP adries v útočnom pakete. Ako bolo v práci vysvetlené, útočníci podvrhnutím maskujú svoju identitu a sťažujú mitigáciu útoku, čo však vedie k vedľajšiemu efektu vo forme spätného rozptylu. Práca sa preto ďalej zaoberala tým, ako je možné použiť spätný rozptyl k detekcii prebiehajúcich útokov. Tento vzťah bol skúmaný pomocou štatistických vlastností spätného rozptylu. Cieľom štatistického prístupu bolo vyhodnotenie závislosti medzi veľkosťou monitorovaného adresového priestoru a počtom pozorovaných spätne rozptýlených paketov. Prirodzene, čím väčší monitorovací priestor máme k dispozícii, tým menšie DDoS útoky je môžeme pozorovať a to navyše s vyššou presnosťou. Táto skutočnosť bola zároveň demonštrovaná na konkrétnych príkladoch rôznych sietí a útokov. V zápätí bolo v práci prezentovaných niekoľko existujúcich prístupov k detekcii DDoS útokov na teleskopoch. Na základe týchto znalosti bola potom navrhnutá vlastná metóda detekcie DDoS útokov schopná detegovať útoky mimo teleskopu v dátach sieťových tokov. Navrhnutá metóda je založená na strojovom učení a vyžaduje anotované dáta, na ktorých je možné natrénovať model rozoznávajúci DDoS útoky. Kľúčovým bodom navrhnutej metódy je teda samotná anotácia dát. Na riešenie problému anotácie bola využitá náhodná povaha spätného rozptylu, ktorá umožňuje pozorovanie prebiehajúceho útoku z rôznych častí internetu. Skupina tokov bola potom v cieľovej sieti, kde sú prítomné aj legitímne zariadenia, označená ako spätný rozptyl v prípade, ak bolo možné korelovať tento výskyt s výskytom spätného rozptylu na nejakom teleskope. Navrhnutá metóda teda vyžaduje použite teleskopu ale len v rámci procesu tvorby sady a pri nasadení sa

zaobíde bez neho. Vyššie popísaná metóda bola následne implementovaná v rámci prúdového systému spracovania tokov a zároveň frameworku NEMEA. Výsledná implementácia bola rozdelená do dvoch modulov. Prvým implementovaným modulom je modul extrakcie rysov. Úlohou tohto modulu je transformovanie skupiny súvisiacich potenciálne spätne rozptýlených tokov do podoby vektoru rysov vhodného pre strojového učenie. Tento proces bol implementovaný principiálne podobne ako agregácia paketov do tokov na NetFlow sonde s použitím pasívneho a aktívneho časovača. Vytvorené vektory rysov, reprezentujúce skupiny tokov, potom smerujú do druhého modulu, ktorý na základe natrénovaného modelu strojového učenia rozhodne o tom, či vektor reprezentuje DDoS útok.

Implementovaná metóda bola vyhodnotená na reálnych dátach tokov organizácie CESNET. Za účelom anotácie týchto dát bol použitý pomerne veľký teleskop organizácie CAIDA. Výsledná anotovaná dátová sada obsahovala 4 súvislé dni dát tokov reprezentovaných v podobe množiny vektorov rysov, pričom prvé dva dni tejto sady boli použité na trénovanie, nasledujúci deň na validáciu a posledný deň na vyhodnotenie modelu. V rámci procesu trénovania bolo vyhodnotených niekoľko rôznych klasifikátorov. Najlepšiemu z týchto klasifikátorov sa podarilo dosiahnuť hodnotu F1 skóre 95,8 %, PPV 95,4 % a citlivosti 96,1 % na protokole TCP, respektíve F1 skóre 96,2 %, PPV 94,6 % a citlivosť 97,9% na protokole ICMP. V závere vyhodnotenia metódy boli taktiež diskutované možnosti ďalšieho pokračovania práce. Z pohľadu zlepšenia modelu sa ponúkajú dva riešenia. Prvou a realizačne jednoduchšou možnosťou je rozšírenie dátovej sady, tak aby pokrývala dlhší časový úsek. Druhá možnosť spočíva v pridaní nových rysov, táto voľba však vyžaduje dodatočnú analýzu a zmenu implementovaných modulov.

Literatúra

[1] Blenn, N., Ghiëtte, V. a Doerr, C. Quantifying the Spectrum of Denial-of-Service Attacks through Internet Backscatter. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security*. ACM, August 2017. DOI: 10.1145/3098954.3098985. Dostupné z: https://doi.org/10.1145/3098954.3098985.

[2] Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*. Association for Computing Machinery (ACM). Júl 1970, zv. 13, č. 7, s. 422–426. DOI: 10.1145/362686.362692. Dostupné z: https://doi.org/10.1145/362686.362692.

[3] *The CAIDA UCSD Network Telescope Aggregated Flow Dataset* [online]. Október 2020 [cit. 2020-11-22]. Dostupné z: https://www.caida.org/data/passive/telescope-flowtuple.xml.

[4] *The CAIDA UCSD Network Telescope Daily Randomly and Uniformly Spoofed Denial-of-Service (RSDoS) Attack Metadata* [online]. Október 2020 [cit. 2020-11-22]. Dostupné z: https://www.caida.org/data/passive/telescope-daily-rsdos.xml.

[5] *The UCSD Network Telescope* [online]. Október 2020 [cit. 2020-11-29]. Dostupné z: https://www.caida.org/projects/network_telescope/.

[6] Carl, G., Kesidis, G., Brooks, R. a Rai, S. Denial-of-service attack-detection techniques. *IEEE Internet Computing*. Institute of Electrical and Electronics Engineers (IEEE). Január 2006, zv. 10, č. 1, s. 82–89. DOI: 10.1109/mic.2006.5. Dostupné z: https://doi.org/10.1109/mic.2006.5.

[7] Cejka, T., Bartos, V., Svepes, M., Rosa, Z. a Kubatova, H. NEMEA: A framework for network traffic analysis. In: *2016 12th International Conference on Network and Service Management (CNSM)*. IEEE, Október 2016. DOI: 10.1109/cnsm.2016.7818417. Dostupné z: https://doi.org/10.1109/cnsm.2016.7818417.

[8] *Topologie sítě CESNET2* [online]. Február 2020 [cit. 2021-05-03]. Dostupné z: https://www.cesnet.cz/sluzby/pripojeni/topologie/.

[9] Cormode, G. a Hadjieleftheriou, M. Methods for finding frequent items in data streams. *The VLDB Journal*. Springer Science and Business Media LLC. December 2009, zv. 19, č. 1, s. 3–20. DOI: 10.1007/s00778-009-0172-z. Dostupné z: https://doi.org/10.1007/s00778-009-0172-z.

[10] Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R. et al. Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and

IPFIX. *IEEE Communications Surveys & Tutorials*. Institute of Electrical and Electronics Engineers (IEEE). Máj 2014, zv. 16, č. 4, s. 2037–2064. DOI: 10.1109/comst.2014.2321898. Dostupné z: https://doi.org/10.1109/comst.2014.2321898.

[11] Jonker, M., King, A., Krupp, J., Rossow, C., Sperotto, A. et al. Millions of targets under attack. In: *Proceedings of the 2017 Internet Measurement Conference*. ACM, November 2017. DOI: 10.1145/3131365.3131383. Dostupné z: https://doi.org/10.1145/3131365.3131383.

[12] Krämer, L., Krupp, J., Makita, D., Nishizoe, T., Koide, T. et al. AmpPot: Monitoring and Defending Against Amplification DDoS Attacks. In: *Research in Attacks, Intrusions, and Defenses*. Springer International Publishing, December 2015, s. 615–636. DOI: 10.1007/978-3-319-26362-5_28. ISBN 978-3-319-26361-8. Dostupné z: https://doi.org/10.1007/978-3-319-26362-5_28.

[13] Luckie, M., Beverly, R., Koga, R., Keys, K., Kroll, J. A. et al. Network Hygiene, Incentives, and Regulation. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, November 2019. DOI: 10.1145/3319535.3354232. Dostupné z: https://doi.org/10.1145/3319535.3354232.

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

[15] Mayfield, P. *Understanding Binomial Confidence Intervals* [online]. [cit. 2020-11-29]. Dostupné z: https://sigmazone.com/binomial-confidence-intervals/.

[16] Moore, D., Shannon, C., Voelker, G. a Savage, S. *Network Telescopes: Technical Report* [online]. Center for Applied Internet Data Analysis (CAIDA), júl 2004. Dostupné z: https://catalog.caida.org/details/paper/2004_tr_2004_04.

[17] Moore, D., Shannon, C., Brown, D. J., Voelker, G. M. a Savage, S. Inferring Internet denial-of-service activity. *ACM Transactions on Computer Systems*. Association for Computing Machinery (ACM). Máj 2006, zv. 24, č. 2, s. 115–139. DOI: 10.1145/1132026.1132027. Dostupné z: https://doi.org/10.1145/1132026.1132027.

[18] Furutani, N., Ban, T., Nakazato, J., Shimamura, J., Kitazono, J. et al. Detection of DDoS Backscatter Based on Traffic Features of Darknet TCP Packets. In: *2014 Ninth Asia Joint Conference on Information Security*. IEEE, September 2014. DOI: 10.1109/asiajcis.2014.23. Dostupné z: https://doi.org/10.1109/asiajcis.2014.23.

[19] Ali, S. H. A., Furutani, N., Ozawa, S., Nakazato, J., Ban, T. et al. Distributed Denial of Service (DDoS) Backscatter Detection System Using Resource Allocating Network with Data Selection. *Memoirs of the Graduate Schools of Engineering and System Informatics Kobe University*. Terrapub. Jún 2015, č. 7, s. 8–13. DOI: 10.5047/gseku.e.2015.001. Dostupné z: https://doi.org/10.5047/gseku.e.2015.001.

[20] Skrjanc, I., Ozawa, S., Dovzan, D., Tao, B., Nakazato, J. et al. Evolving cauchy possibilistic clustering and its application to large-scale cyberattack monitoring. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*.

IEEE, November 2017. DOI: 10.1109/ssci.2017.8285203. Dostupné z: https://doi.org/10.1109/ssci.2017.8285203.

[21] *DDoS Attacks* [online]. [cit. 2020-12-26]. Dostupné z: https://www.imperva.com/learn/ddos/ddos-attacks/.

[22] *Flowmon Collector* [online]. [cit. 2020-12-05]. Dostupné z: https://www.flowmon.com/en/products/appliances/netflow-collector.

[23] *NTP Amplification* [online]. [cit. 2020-12-28]. Dostupné z: https://www.imperva.com/learn/ddos/ntp-amplification/.

[24] *Ping flood (ICMP flood)* [online]. [cit. 2020-12-28]. Dostupné z: https://www.imperva.com/learn/ddos/ping-icmp-flood/.

[25] *SYN Flood Attack* [online]. [cit. 2020-12-28]. Dostupné z: https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/.

[26] *UDP Flood* [online]. [cit. 2020-12-28]. Dostupné z: https://www.imperva.com/learn/ddos/udp-flood/.

[27] Listvan, R. *Introducing flow formats and their differences* [online]. December 2018 [cit. 2020-12-05]. Dostupné z: https://www.flowmon.com/en/blog/introducing-flow-formats-and-their-differences.

[28] *Threat Alert: TCP Amplification Attacks* [online]. November 2019 [cit. 2020-12-28]. Dostupné z: https://blog.radware.com/security/2019/11/threat-alert-tcp-reflection-attacks/.

[29] Kupreev, O., Gutnikov, A. a Badovskaya, E. *DDoS attacks in Q3 2020* [online]. Október 2020 [cit. 2020-12-26]. Dostupné z: https://securelist.com/ddos-attacks-in-q3-2020/99171/.

[30] Cox, J. *NetFlow vs IPFIX* [online]. Máj 2020 [cit. 2020-12-05]. Dostupné z: https://www.ittsystems.com/netflow-vs-ipfix/.

[31] Ferguson, P. a Senie, D. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing* [online]. Máj 2000 [cit. 2020-12-28]. Dostupné z: http://www.rfc-editor.org/rfc/rfc2827.txt.

[32] Claise, B. *Cisco Systems NetFlow Services Export Version 9* [online]. Október 2004 [cit. 2020-12-05]. Dostupné z: http://www.rfc-editor.org/rfc/rfc3954.txt.

[33] Trammell, B., Boschi, E., Mark, L., Zseby, T. a Wagner, A. *Specification of the IP Flow Information Export (IPFIX) File Format* [online]. Október 2009 [cit. 2020-12-01]. Dostupné z: https://www.ietf.org/rfc/rfc5655.txt.

[34] Trammell, B., Wagner, A. a Claise, B. *Flow Aggregation for the IP Flow Information Export (IPFIX) Protocol* [online]. September 2013 [cit. 2020-12-01]. Dostupné z: https://www.ietf.org/rfc/rfc7015.txt.

[35] Claise, B., Trammell, B. a Aitken, P. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information* [online]. September 2013 [cit. 2020-12-01]. Dostupné z: http://www.rfc-editor.org/rfc/rfc7011.txt.

[36] Schilling, M. F. The Surprising Predictability of Long Runs. *Mathematics Magazine*. Informa UK Limited. Apríl 2012, zv. 85, č. 2, s. 141–149. DOI: 10.4169/math.mag.85.2.141. Dostupné z: https://doi.org/10.4169/math.mag.85.2.141.

[37] Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. et al. An Overview of IP Flow-Based Intrusion Detection. *IEEE Communications Surveys & Tutorials*. Institute of Electrical and Electronics Engineers (IEEE). Apríl 2010, zv. 12, č. 3, s. 343–356. DOI: 10.1109/surv.2010.032210.00054. Dostupné z: https://doi.org/10.1109/surv.2010.032210.00054.

Príloha A

Vybraná množina rysov

Rys	Dôležitosť rysu v %
Počet unikátnych cieľových podsietí s prefixom siete /24	71,3
Počet unikátnych cieľových podsietí s prefixom siete /24 normalizovaný počtom tokov	14,2
Počet unikátnych zdrojových portov	3,5
Počet unikátnych cieľových IP adries normalizovaný počtom tokov	2,4
Počet unikátnych zdrojových portov normalizovaný počtom tokov	2,2
Počet unikátnych cieľových portov normalizovaný počtom tokov	1,7
Priemerný počet tokov za sekundu	1,4
Počet unikátnych cieľových portov	1,2
Priemerný počet paketov na tok	0,9
Maximálny počet tokov za minútu	0,7
Počet unikátnych cieľových IP adries	0,4
Počet tokov	0,1

Tabuľka A.1: Vybrané rysy protokolu TCP spolu s ich dôležitosťou vo finálnom modele strojového učenia (`feature_importances_`).

Rys	Dôležitosť rysu v %
Počet unikátnych cieľových IP adries normalizovaný počtom tokov	76,5
Štandardná odchýlka počtu paketov na tok	17,2
Počet unikátnych cieľových podsietí s prefixom siete /24 normalizovaný počtom tokov	1,7
Počet unikátnych cieľových podsietí s prefixom siete /24	1,6
Štandardná odchýlka počtu bajtov na paket	0,8
Priemerný počet bajtov na paket	0,8
Priemerný počet tokov za sekundu	0,7
Priemerný počet paketov na tok	0,4
Maximálny počet tokov za minútu	0,3

Tabuľka A.2: Vybrané rysy protokolu ICMP spolu s ich dôležitosťou vo finálnom modele strojového učenia (`feature_importances_`).

Príloha B

Obsah priloženého DVD

Na priloženom DVD sa nachádzajú nasledovné súbory a zložky:

thesis/ text diplomovej práce a jeho zdrojové kódy

- **xmarus07_spatny_rozptyl.pdf** text diplomovej práce formát PDF
- **src/** zdrojové kódy textu diplomovej práce vo formáte LaTeX

src/ implementácia NEMEA modulov spolu s trénovaním modelu strojového učenia

- **README.md** manuál k spusteniu modulov a experimentov
- **backscatter/** implementácia modulu extrakcie rysov
- **backscatter_classifier/** implementácia modulu klasifikácie
- **experiments/** experimenty
  - **probability.ipynb** Jupyter zošit obsahujúci štatistickú analýzu spätného rozptylu
  - **dataset_annotation.ipynb** Jupyter zošit obsahujúci proces anotácie a filtrovania dátovej sady udalostí
  - **training.ipynb** Jupyter zošit obsahujúci proces trénovania modelu strojového učenia
- **datasets/** anonymizovaná dátová sada udalostí s a bez simulovaných prerušení

Priložené médium neobsahuje vzhľadom k veľkosti a citlivej povahe dáta tokov organizácie CESNET a dáta z teleskopu organizácie CAIDA.