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

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

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


# ODVOZOVÁNÍ VZORU PRO MITIGACI DDOS ÚTOKŮ

**INFERENCE OF DDOS MITIGATION RULES**

**BAKALÁŘSKÁ PRÁCE**
**BACHELOR'S THESIS**


**AUTOR PRÁCE** **ERIK BELKO**
**AUTHOR**

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



**BRNO 2022**

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

Ústav počítačových systémů (UPSY) Akademický rok 2021/2022

# Zadání bakalářské práce


Student: **Belko Erik**
Program: Informační technologie
Název: **Odvozování vzoru pro mitigaci DDoS útoků**
**Inference of DDoS Mitigation Rules**
Kategorie: Počítačové sítě

Zadání:

1. Nastudujte dostupnou literaturu o DDoS útocích a jejich typech.
2. Nastudujte dostupnou literaturu o technikách mitigace objemových DDoS útoků. Zaměřte se na techniky využívající strojové učení.
3. Navrhněte metodu pro odvození vzoru v paketech, které jsou majoritně zastoupeny v DDoS útoku.
4. Navrženou metodu implementujte.
5. Implementaci vyhodnoťte simulací v laboratorním prostředí.
6. Diskutujte dosažené výsledky a navrhněte další rozšíření.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 až 3 zadání.

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 2021
Datum odevzdání: 11. května 2022
Datum schválení: 29. října 2021

# Abstrakt

Táto práca sa zaoberá útokmi DDoS, ich konkrétnymi typmi a spôsobom ich mitigácie. Cieľom tejto práce je navrhnúť metódu pre odvodenie vzoru z dát paketu pre následnú mitigáciu DDoS útoku a implementovať ju. Zvolená metóda využíva na odvodenie vzoru rozdeľovanie dát paketu na N-gramy. Metóda využíva vzorky s dátami zachytenými počas legitímnej prevádzky a počas DDoS útoku. V práci sú taktiež popísané ďalšie navrhované metódy a so zvolenou metódou sú nad dátami rôznych veľkostí vykonané experimenty.



# Abstract

This thesis deals with DDoS attacks, their specific types and ways of mitigating them. The aim of the thesis is to propose a method for inferring a pattern from a packet payload for subsequent DDoS attack mitigation and implement it. The chosen method uses the partitioning of the packet payload into N-grams to infer the pattern. The method utilizes samples with data captured during legitimate traffic and during a DDoS attack. Other proposed methods are also described in the thesis and experiments are performed with the selected method over data of different sizes.



# Kľúčové slová

DDoS útok, mitigácia DDoS, dáta paketu, odvodenie vzoru, N-gramy, sieť

# Keywords

DDoS attack, DDoS mitigation, packet payload, pattern inference, N-grams, network



# Citácia

BELKO, Erik. *Odvozování vzoru pro mitigaci DDoS útoků*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Žádník, Ph.D.

# Odvozování vzoru pro mitigaci DDoS útoků

## Prehlásenie

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

.....
Erik Belko
16. mája 2022


## Poďakovanie

Ďakujem pánovi Ing. Martinovi Žádníkovi, Ph.D., za vedenie tejto práce, za cenné rady, odbornú pomoc, ľudský prístup, trpezlivosť a veľkú ochotu pri konzultáciách.

# Obsah

# Kapitola 1

# Úvod

Spolu s rozvojom informačných technológii predstavuje internet v dnešnej dobe každodennú neoddeliteľnú súčasť spoločnosti. Internet je nástroj, ktorý predstavuje a prináša veľké množstvo výhod, no spolu s výhodami prináša aj množstvo negatívnych vecí a bezpečnostných rizík. Medzi riziká patrí veľké množstvo rôznych kyberútokov. Zámerom týchto útokov je poškodiť firmu spoločnosť alebo štátny orgán. Jedným z možných cieľov útoku je poškodiť obeť tým, že jej užívateľom, častokrát zákazníkom ktorí za službu platia, je zamedzený prístup k službe. Tento typ útoku často prináša nepríjemnosti zákazníkom a finančné straty obetiam.

Práve útok blokovaním a odopretím služby, serveru alebo siete, takzvaný Denial of Service (DoS), je populárny typ útoku medzi útočníkmi. Hlavne jeho distribuovaný variant Distributed Denial of Service (DDoS), ak útok pochádza z viacerých zdrojov naraz. DDoS útok bude vysvetlený v časti 2.2. DDoS útok je distribuovaný v takzvaných botnetoch, množstve infikovaných zariadeniach, ktoré útočník zneužíva na útok. Tento útok teda zneprístupňuje obeť útoku zaplavením veľkého množstva prevádzky. DDoS útoky patria k najbežnejším útokom a ochrana pred nimi nie je jednoduchá a dá sa povedať, že útočníci sú vždy o krok vpred. Pristupuje sa k nej rôznymi prístupmi a je témou mnoho výskumov v oblasti bezpečnosti sietí. Jedným s prístupov je ochrana pomocou blokovania zdrojov. Tá nie je jednoduchá, keďže prevádzka je rozdistribuovaná a prichádza z viacerých zdrojov. Ďalším dôvodom kedy sa nejaví blokovanie na základe zdrojov ako efektívne je falšovanie zdrojových IP adries útočníkmi. Preto sa zámery výskumov presunuli aj na iné spôsoby. Jednému zo spôsobov sa venuje aj táto práca, a to odvodenie vzoru, ktorý je majoritne zastúpený v DDoS útokoch.

Hlavným cieľom tejto práce je navrhnúť metódu pre odvodenie vzoru z dátového obsahu paketov, ktorý je majoritne zastúpený v DDoS útokoch. Metóda by mala byť schopná odvodiť vzor v útočiacej prevádzke, ktorý čo najviac charakterizuje pakety patriace útoku. Odvodený vzor by mal slúžiť a pomôcť pri následnej mitigácií zmierniť prebiehajúci útok. Cieľom tejto práce zároveň ale nie je navrhnúť metódu, ktorá by prispela k blokovaniu všetkých typov DDoS útokov, ale k blokovaniu takých typov DDoS útokov, ktoré nejdú blokovať inak ako na základe odvodeného vzoru z dátový obsahu paketu. Je to teda jedna z mnohých metód blokovania DDoS útokov. V rámci tejto práce bolo zvažovaných a navrhovaných viacero možných metód a následne bola vybraná jedna, ktorá bola implementovaná a pomocou ktorej boli vykonané experimenty.

Táto práca je štrukturovaná do viacerých kapitol. Kapitola 2 bude vysvetľovať potrebné informácie ohľadne DDoS útokov, ich typoch a využitia botnetu. V tejto kapitole bude zároveň spomenutá aj mitigácia DDoS útokov a vysvetlené budú aj niektoré spôsoby mitigácie.

Zároveň bude v kapitole vysvetlené aj využitie falšovania IP adries útočníkmi a na záver bude v kapitole vysvetlený všeobecný princíp n-gramov, ktoré budú využité v navrhnutej metóde. V kapitole 3 bude vysvetlená zvolená metóda, zároveň s objasnením cieľa tejto práce a spomenuté budú aj ďalšie zvažované metódy. Implementácia a použité technológie zvolenej metódy budú popísané v kapitole 4 a následne vykonané experimenty s touto metódou bude zahŕňať kapitola 5.

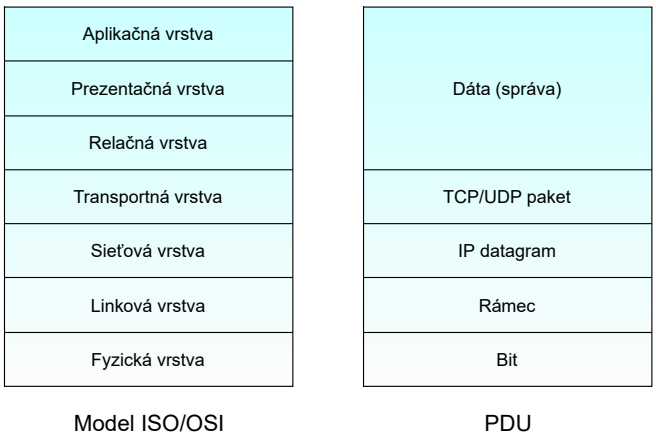
# Kapitola 2

# Teória

## 2.1 ISO/OSI model

V tejto práci bude viackrát spomínaný dátový obsah paketu, takzvaný payload, s ktorým práca pracuje, preto je vhodné definovať tento pojem a čo sa ním myslí v rámci tejto práce. Pojem payload bude vysvetlený v tejto časti, zároveň aby sme mohli správne definovať payload potrebujeme využiť už existujúcich definícií rámcov, paketov, datagramov, a vrstiev na ktorých sú tieto základné dátové jednotky, takzvané PDU. Preto je v rámci tejto časti práce vysvetlený aj vrstvový model počítačových sietí ISO/OSI, ktorý sa používa pre popis sieťovej architektúry. Poznatky popísané v tejto časti boli inšpirované knihou [18] a [6].

Model ISO/OSI sa dá považovať za hlavnú architektúru počítačových sietí, podľa ktorej sa skladá veľa dnešných protokolov. Model ISO/OSI je tvorený zo sedem vrstiev. Od najnižšej to sú: fyzická, linková, sieťová, transportná, relačná, prezentačná, aplikačná. Každá vrstva definuje služby, protokoly a funkcie pre prenos dát na rovnakej vrstve. Pri komunikácií vrstvy využívajú služby nižších vrstiev bez toho aby potrebovali poznať správanie tejto vrstvy a spôsob akým bola implementovaná. Na jednotlivých vrstvách sú definované už spomínané základné dátové jednotky, takzvané PDU (Process Data Unit). Znázornené sú na obrázku 2.1, ktorý ukazuje jednotlivé vrstvy modelu ISO/OSI a zároveň aj PDU v jednotlivých vrstvách. PDU nejakej vrstvy je PDU vyššej vrstvy, spolu s pridanou príslušnou hlavičkou danej vrstvy.




Obr. 2.1: Základné dátové jednotky (PDU) v jednotlivých vrstvách modelu ISO/OSI inšpirované obrázkom 1.6 z [18]

Na základe jednotlivých PDU daných vrstiev, definícia payloadu paketu v rámci tejto práce je nasledujúca:

- **Payload** paketu sa označuje ako dáta prenášané TCP alebo UDP paketom, sú to dáta ktoré posiela odosielateľ príjemcovi. Payload je teda obsah TCP alebo UDP paketu (PDU štvrtej vrstvy), do payloadu sa nepočíta hlavička TCP alebo UDP paketu. Payload na obrázku 2.1 je PDU nazvané dáta (správa), a nachádza sa teda na najvyšších troch vrstvách. Veľkosť payloadu sa môže líšiť.

## 2.2 DDoS útok

V nasledujúcej časti tejto práce bude vysvetlený DDoS útok, jeho vlastnosti a ako vzniká. Súčasťou je aj vysvetlenie pojmu botnet. Následne v 2.2.2 budú uvedené a vysvetlené typy útokov DDoS, ktoré sú relevantné pre túto prácu.

DDoS útok je hrozba ktorá narúša bezpečnosť internetu. Je cieľom veľkého počtu výskumov, ktoré sa snažia nájsť spôsob ako sa tejto hrozbe brániť. DDoS útok, celým názvom Distributed Denial of Service, je úmyselný pokus narušiť normálnu prevádzku cieľového servera, služby alebo nejakej siete zahltením cieľa záplavou internetovej prevádzky. Hlavnou motiváciou DDoS útoku teda nie je ovládnuť cieľ, alebo získať určité dáta, ale znefunkčniť a zneprístupniť cieľovú službu ostatným užívateľom, takzvanej legitímnej prevádzke [2]. Užívatelia služby si často platia za tieto služby a preto je ich vyradenie DDoS útokom pre poskytovateľa nežiadúce a veľmi nákladné. Neraz bol tento typ použitý na ciele ako vládne servery alebo servery väčších spoločností. DDoS útok je ale využívaný veľmi často, kvôli jeho cene a dostupnosti. Aj keď je DDoS útok pomerne lacný, vie narobiť obrovské finančné škody cieľu. Zahltenie cieľa môže trvať časovú dobu alebo neurčitý čas. Útoky DDoS využíva ako zdroje útočiacej prevádzky viac počítačových systémov, takzvaný Botnet 2.2.1 mechanizmus. Využívané sú nielen počítače ale aj ďalšie sieťové zdroje, často zariadenia IoT, takzvaný Internet of Things [20].
Rozlíšiť a oddeliť legitímnu prevádzku od útočiacej prevádzky spôsobenej DDoS útokom je náročné a pristupuje sa k tomuto problému rôzne. Niekedy sa stane, že aj väčšie množstvo legitímnej prevádzky môže indikovať bežný symptóm DDoS, ktorým je náhle spomalenie alebo nedostupnosť stránky alebo služby. Nástroje na analýzu návštevnosti môžu pomôcť rozpoznať niektoré zo signálov DDoS útoku, ktorými môžu byť napríklad:

- množstvo požiadaviek pochádzajúcich z určitého rozsahu adries
- záplava návštevnosti s rovnakým vzorom správania, ako je typ zariadenia alebo verzia webového prehliadača
- výkyvy v určitých hodinách dňa alebo vzory, ktoré sa zdajú byť neprirodzené

Samozrejme príznakov DDoS útoku je viac, niektoré sú už špecifickejšie a odlišujú sa od seba pri daných typoch 2.2.2 DDoS útoku [27].

Pri obrane pred DDoS útokmi sa nesmie zabúdať na prevenciu, ktorá je jej podstatnou súčasťou. Útočníci hľadajú nové spôsoby akými možno zaútočiť a prispôsobujú sa prípadným zákrokom proti nim. Motivácia môže byť rôzna. Na obranu DDoS, sú používané mitigačné techniky, popísané v časti 2.3, ktoré sú vyvinuté tak aby si poradili s rôznymi typmi útokov. Avšak útočníci sa snažia obchádzať mitigáciu DDoS, napríklad aj sfalšovaním zdrojovej IP adresy paketu, čím zakrývajú svoju totožnosť, tento princíp bude popísaný

v časti 2.4. Prípadne útočníci útočia na celú podsieť. Týmito spôsobmi sa snažia aby nebolo možné prevádzku odfiltrovať a zablokovať na úrovni tretej vrstvy modelu OSI, teda IP protokolu. V takom prípade môže byť užitočné filtrovanie na základe odvodeného vzoru z payloadu paketov. Odvodenie tohto vzoru je zároveň motiváciou tejto práce.

### 2.2.1 Botnet

Rozdiel medzi DoS útokom a DDoS útokom je okrem iného taký, že DDoS využíva na zaútočenie na cieľ botnet. DoS je útok systému na systém, čiže jeden počítač alebo stroj útočí na cieľ a snaží sa ho zaplaviť návštevnosťou a tým sa zdroj stane nedostupným. Pritom ak sa jedná o DDoS útok, ten je distribuovaný medzi viacero počítačov alebo strojov, z ktorých útok pochádza. Tieto stroje sú infikované a pripravené zaútočiť spoločne a naraz, čím je zaplavenie cieľa masívnejšie, rýchlejšie a horšie sa detekuje pôvod útoku, keďže útok prichádza z viacerých zdrojov [14].

Botnet je mocný nástroj, ktorý sa používa na kybernetické zločiny. Tými môžu byť útoky ako napríklad takzvaný Click Fraud alebo takzvaný Spam Email. No najčastejšie využitie botnetu je práve pri DDoS útokoch [10]. Ako sa spomína v [19], botnet pozostáva z veľkého množstva infikovaných zariadení v sieti, takzvaní bots alebo zombies, ktoré sú na diaľku ovládané útočníkom, ktorý určuje na aký cieľ a kedy zaútočiť. Útok pomocou botnetu sa dá zvyčajne charakterizovať tromi vlastnosťami, ktorými sú podobnosť zdrojov útoku, divergencia medzi normálnym a útočným sieťovým tokom a automatizácia vykonávania útoku.

Bot, zariadenie ktoré je infikované a pripravené na pokyn k útoku, je prepojené k ďalším botom pomocou internetu, spolu tvoria botnet. Botnet teda môže pozostávať z niekoľko tisíc zariadení. Tieto zariadenia nepredstavujú iba infikované počítače ale v dnešnej dobe sú čoraz častejšie zahrnuté v botnetoch aj zariadenia Internet of Things (IoT). Počet IoT zariadení rapídne stúpa, a keďže tieto zariadenia sú vyrobené bez ochrany a často sú na nich používané predvolené heslá, tak sú vhodnými botmi. Útočníci okrem toho, že môžu získať botnet skenovaním Internetu a infikovaním vhodných zariadení, si môžu botnet kúpiť alebo prenajať od operátorov za nie vysoké ceny [24].

Botnet [10] je ovládaný útočníkom takzvaným bot herderom alebo botmasterom. Botmaster pošle príkaz s inštrukciami botom pomocou zariadenia na to určeného, takzvaný command and control server. Tento postup zabezpečuje, že pôvodné zariadenie útočníka je ťažko vystopovateľné.

### 2.2.2 DDoS - typy

Typov DDoS útokov je veľmi veľký počet, preto je ťažké ich kategorizovať. Predsa len je v odvetví uznávané rozdelenie do troch základných kategórií, poznáme teda útoky na protokol, útoky na aplikačnú vrstvu a volumetrické útoky. Medzi týmito kategóriami ale nastáva aj určité prekrývanie [24].

V nasledujúcej časti tejto práce budú vymenované a vysvetlené spomínané tri kategórie útokov a následne konkrétne typy útokov, podľa [24, 27, 20]. Keďže typov je veľa, spomenuté budú iba niektoré, príklady pre každú spomínanú kategóriu a hlavne tie typy, ktoré sú relevantné pre túto prácu, najmä tie ktoré využívajú falšovanie IP adries. Pri sfalšovaní zdrojovej IP adresy útočník zakrýva svoju totožnosť, jedná sa o takzvaný IP spoofing, ktorý bude stručne vysvetlený v časti 2.4. Kvôli sfalšovaniu IP adresy sa stáva filtrovanie na základe zdrojovej IP adresy neefektívnym obranným mechanizmom a alternatívou môže byť filtrovanie na základe vzorov v payloade paketu, ktoré je zároveň motiváciou tejto práce.

### Útoky na protokol

Pri tejto kategórii, je dôležité spomenúť že sa nezameriavajú na zdroje vyššej úrovne, webový server a podobne, ale zameriavajú sa na slabé miesta protokolov a ich bežné správanie. Väčšinou sa jedná o protokoly z tretej a štvrtej vrstvy modelu OSI (ICMP, UDP, TCP, ...). Konkrétnym cieľom je vyčerpať nadmernou spotrebou zdroje sieťového vybavenia ako firewally a vyrovnávače zaťaženia, alebo výpočtové možnosti siete, miera pri týchto útokoch sa teda udáva v paketoch za sekundu. Príkladom útoku z tejto kategórie, môže byť napríklad takzvaný útok SYN flood, spomenutý v 2.2.2.

### Útoky na aplikačnú vrstvu

Táto kategória útokov, ako z názvu vyplýva sa zameriava na aplikačnú vrstvu modelu OSI, webové servery, platformy webových aplikácií. Aplikačná vrstva je najvyššou, siedmou vrstvou, preto sa táto kategória niekedy označuje aj ako takzvaný layer 7 DDoS attack. Cieľom útoku je vyčerpať zdroje serveru, na ktorom sa generujú webové stránky a sú doručované v odpovedi na HTTP požiadavky, a zneprístupniť službu používateľom. Vytváranie odpovedí na požiadavky je pre server náročné keďže musí pracovať s databázou a načítavať často viacero súborov, aby sprístupnil webovú stránku. Pri tejto kategórií je náročné odhaliť útočiacu prevádzku medzi legitímnou. Tieto útoky sa okrem zamerania na zraniteľnosť aplikácií, môžu zamerať aj na zneužitie protokolov siedmej vrstvy (HTTP, HTTPS, SNMP, ...). Miera pri týchto útokoch sa udáva v požiadavkách za sekundu. Príkladom útoku z tejto kategórie, môže byť napríklad takzvaný útok HTTP flood, spomenutý v 2.2.2.

### Volumetrické útoky

Kategória volumetrických útokov sa zameriava na preťaženie a zahltenie spotrebovaním celej šírky pásma, výpočetného výkonu alebo dátovej štruktúry, napríklad pomocou formy amplifikácie alebo iného spôsobu vytvárania masívnej návštevnosti, na to je využívané množstvo vytvorených požiadaviek z útočiaceho botnetu. Práve využitie botnetu je základom dobrého volumetrického útoku. Obrovské množstvo prevádzky a dát je posielané do siete obete, preto sú tieto útoky nazývané aj záplavy. Zahltenie šírky pásma medzi obeťou a internetom spôsobí zamietnutie prístupu legitímnym používateľom.

V nadväznosti na vyššie spomínané kategórie DDoS útokov bude v nasledujúcej časti vysvetlené fungovanie niektorých typov DDoS útokov:

### SYN flood

Zaplavenie paketmi SYN, alebo takzvaný SYN flood je útok pri ktorom je zneužitý spôsob ustanovenia spojenia, takzvaný three-way handshake, ktorý sa využíva na nadviazanie TCP spojenia. Pri ustanovení TCP spojenia klient posiela serveru paket TCP SYN a žiada tak server o nadviazanie spojenia. Server po prijatí paketu odpovedá danému klientovi poslaním paketu TCP SYN-ACK a zaradí požiadavku na pripojenie do frontu požiadaviek. Server následne pri korektnom nadviazaní pripojenia očakáva potvrdenie TCP SYN-ACK paketu, paketom TCP ACK. Ak však príde k útoku, je tento spôsob nadviazania spojenia zneužitý tak, že útočník vytvára množstvo požiadaviek zasielaním TCP SYN paketov, obeť tak musí otvárať množstvo TCP spojení a zaraďovať tieto požiadavky do frontu. Útočník následne tento spôsob ale zneužije nevykonaním tretieho kroku a nikdy nepošle potvrdzujúci TCP

ACK paket. Namiesto toho posiela ďalšie množstvo TCP SYN paketov. Útok spôsobí vyčerpanie zdrojov obete, ktorá nie je schopná prijímať žiadne nové požiadavky na spojenie, ani od legitímnych užívateľov, pretože je jej front zaplnený nepotvrdenými TCP spojeniami.

### HTTP flood

Záplava požiadavkami HTTP, alebo takzvaný HTTP flood je útok pri ktorom sa zneužíva protokol HTTP. Týmto protokolom je umožnená komunikácia klientov s web servermi na internete. Pri nadviazaní spojenia sa používajú požiadavky HTTP GET, na vyžiadanie dát zo serveru, alebo HTTP POST na zaslanie dát na server. Útočník zasiela množstvo týchto požiadaviek na server bez čakania na odpoveď. Dochádza k vyčerpaniu zdrojov servera, ktorý sa snaží odpovedať na všetky požiadavky, a pritom musí pracovať s databázou. Útok je zosilnený použitím botnetu. Pri útokoch dochádza k napodobňovaniu legitímnych požiadaviek HTTP GET alebo HTTP POST, preto je tento útok ťažko odhaliteľný od bežnej prevádzky. Okrem toho HTTP flood má menší objem a nie sú pri nej použité techniky ako sa používajú v nižších sieťových vrstvách, napríklad falšovanie IP adries, takzvaný IP spoofing, vysvetlený v časti 2.4, zlý formát paketov alebo reflexia.

### DNS flood

Služba DNS (Domain Name System) je internetová služba, pomocou ktorej zariadenia vyhľadajú špecifický webový server aby sa dostali k obsahu, tento proces prebieha pomocou prekladu. Keď sa používatelia snažia pripojiť na web stránku, tak DNS server poskytuje preklad bežných ľahko zapamätateľných doménových mien na ťažšie zapamätateľné IP adresy, ktoré im prislúchajú. Pri útoku záplavou DNS požiadaviek, takzvaný DNS flood, posiela útočník veľké množstvo požiadaviek na DNS server obete, s cieľom preťažiť ho. Tak ako aj pri ostatných záplavových útokoch, napríklad SYN flood 2.2.2 alebo HTTP flood 2.2.2, DNS server obete sa snaží odpovedať a vyriešiť všetky prijaté požiadavky. Server odpovedá na všetky, pretože nevie rozoznať útočiace požiadavky medzi legitimnými. Pri úspešnom DDoS útoku je server zahltený a nedokáže reagovať, čím sa webová stránka stáva neprístupnou pre legitímnych užívateľov. DNS flood sa líši od DNS reflexívno-amplifikačných útokov tým, že obeťou je DNS server, naproti tomu pri DNS amplifikačnou útoku je DNS server zneužitý ako zbraň.

### DNS reflexívno-amplifikačný útok

Pri tomto type útoku je hlavným prvkom falšovanie zdrojovej IP adresy, popísané v časti 2.4. Obeťou nie je DNS (Domain Name System) server ako pri DNS záplavách, takzvaných DNS floods, ale práve DNS server je zneužitý ako dôležitý prvok útoku. Útočník pošle požiadavky v pakete so sfalšovanou zdrojovou IP adresou, ktorá patrí obeti, na voľne dostupný DNS server a ten odpovedá. DNS server odpoveď neposiela naspäť útočníkovi, ale už na sfalšovanú IP adresu, ktorou je IP adresa obete. Obeť tým pádom dostane množstvo odpovedí od DNS servera, ktorá avšak neposlala na tento server požiadavky. Okrem útoku na konkrétny cieľ, sa tento útok využíva aj na vyčerpanie šírky pásma siete.

Sila tohto útoku môže byť zvýšená kombináciou s amplifikačnými technikami, príkladom môže byť, že požiadavky na DNS server sú zložitejšie a nepožadujú iba IP adresu daného doménového mena. To výrazne zvyšuje veľkosť odpovedí, útočníci sa snažia o to aby generované odpovede od DNS serveru boli čo najväčšie, čo samozrejme zvyšuje dopad útoku. Útok okrem iného výrazne naberá na sile ak je využitý aj botnet, [25, 9].

#### NTP reflexívno-amplifikačný útok

Ako aj pri DNS reflexívno-amplifikačnom útoku spomenutom vyššie je aj pri tomto type útoku súčasťou falšovanie zdrojovej IP adresy, popísané v 2.4. Pri tomto útoku je zneužitý NTP (Network Time Protocol) server na zahltenie cieľa, či už siete alebo serveru alebo iného zariadenia. NTP server slúži pre prijímanie a odpovedanie na požiadavky na synchronizáciu vnútorných hodín počítačov v sieti. NTP protokol zaisťuje, aby všetky počítače v sieti mali rovnaký a presný čas. NTP server sa dá ale zneužiť na amplifikačný DDoS útok tak, že je na NTP server posielané množstvo požiadaviek so sfalšovanou zdrojovou IP adresou, ktorá patrí obeti. Tým pádom NTP server odpovedá na tieto požiadavky a odpovede posiela cieľu. Amplifikáciou vzniká veľké množstvo UDP prevádzky, ktorá zahltí cieľ a ten sa stane nedostupným, [26].

## 2.3 Mitigácia DDoS útokov

Táto časť práce sa zaoberá a vysvetľuje niektoré vybrané princípy mitigácie DDoS útokov. Poznatky popísané v tejto časti boli inšpirované z [17, 22, 23, 28].

Mitigácia patrí medzi obranné mechanizmy pred DDoS útokmi. Okrem nej medzi dôležité súčasti obranných mechanizmov patrí aj prevencia. Tá je veľmi dôležitá na obranu pred značným počtom DDoS útokov, útoky prichádzajú však stále s novými signatúrami a spôsobmi, kedy samotná prevencia nie je dostatočná. Metódy ochrany by mali mať za úlohu zabrániť odstaveniu služby, čo je často náročná úloha ku ktorej sa pristupuje rôznymi spôsobmi. Okrem obrany služby je dôležité popri úspešnej obrane nezabrániť bežnej legitímnej prevádzke pristúpiť k službe. Tento prípad je nežiadúci, napriek tomu nastáva často a ťažšie sa bráni, pretože útočníci sa snažia aby vlastnosti prevádzky vytvorenej nimi boli podobné tej legitímnej, a tým neboli ľahko odhalitelné. Zabránenie bežnej legitímnej prevádzke vlastne znamená pomoc pre útočníkov. Veľkou výzvou pre ochranu je teda malá miera falošne pozítívne označených paketov, ktoré patria legitímnej prevádzke, teda takzvaných false positives. V nasledujúcej časti budú popísané, mitigácia a vybrané spôsoby mitigácie, ktoré sú relevantné pre túto prácu.

Jedným z jednoznačných identifikátorov počas DDoS útoku je spomalenie služby, neskôr dochádza až k jej vyradeniu. Dosiahnutím celej šírky pásma siete alebo preťažením služby môže tento problém dosiahnuť, respektíve zapríčiniť aj bežná legitímna prevádzka. Z tejto príčiny je dôležité podrobné skúmanie okolností. Medzi podozrivé patrí nadmerné podobné požiadavky z rovnakej skupiny IP adries alebo dokonca z jednej konkrétnej IP adresy. Pri mitigácii útoku je jednou z úloh rozhodnúť v ktorom bode útok mitigovať. Pri útokoch, ktoré útočia na sieť, zahlcujú sieťovú šírku je lepšie útok mitigovať čo najbližšie k pôvodu útoku ako sa dá. V ďalšom prípade, napríklad pri útoku na server a jeho prostriedky, by mali byť mitigačné techniky zabezpečené na strane obeti. Mitigácia sa často vzťahuje na konkrétne sieťové zariadenie, príkladom môže byť firewall. Postup mitigácie by sa dal zhrnúť do troch krokov, ktorými sú, presmerovaniu prevádzky cez sieť bezpečnostnej spoločnosti alebo zariadenia ktoré vykonáva mitigačné a filtračné techniky, filtrovanie prevádzky podľa rôznych filtračných techník, a poslanie bežnej legitímnej prevádzky naspäť aby dosiahla cieľovej adresy.

Na základe motivácie tejto práce budú ďalej spomenuté iba niektoré konkrétne typy mitigácie, ktoré sú relevantné pre túto prácu.

### Blokovanie na základe IP adries

Táto metóda mitigácie obmedzuje prístup niektorým IP adresám. Zavádza sa takzvaný blacklisting IP adries. Blacklisting IP adries je mechanizmus, ktorým je kontrolovaný prístup. Prístup je povolený na základe preddefinovanej tabuľky, ktorá sa nazýva čierna listina, takzvaný blacklist. V tejto listine sa nachádzajú všetky IP adresy, ktorých prístup bol zamietnutý. Tento mechanizmus teda povoľuje prístup všetkým IP adresám okrem tých, ktoré sa nachádzajú na čiernej listine. Povolené IP adresy, ktorých prístup sa zamietne až v neskoršej fáze, sa pridajú medzi zamietnuté a čierna listina sa aktualizuje. Problémom a nedostatkom tejto mitigačnej metódy je falšovanie IP adries. Útočníci po tom ako bola nimi používaná IP adresa pridaná na čiernu listinu, jednoducho použijú inú, ktorá ešte nie je medzi zakázanými adresami. Táto práca skúma alternatívu, keď blokovanie na základe IP adries nie je možné kvôli falšovaniu zdrojových IP adries. Falšovanie IP adries bude vysvetlené v časti 2.4.

### Blokovanie na základe firewall pravidiel

Mitigácia útokov môže byť zabezpečená aj pomocou firewallu. Firewall je buď špeciálne hardvérové zariadenie alebo softvér na určitom zariadení, ktorým môže byť napríklad smerovač alebo aj počítač. Firewall rozdeľuje dôveryhodnú, privátnu, sieť od nedôveryhodnej, vonkajšej, siete. Zvyčajne býva umiestnený medzi nimi, a zabezpečuje ochranu pred viacerými nebezpečenstvami. Vďaka svojej pozícii na okraji siete sa od firewallu vyžaduje mnoho ďalších úloh než len blokovanie sieťových pripojení, napríklad vykonávanie NAT, ukončovanie VPN, ochrana pred vírusmi a škodlivým softvérom a dokonca aj ochrana pred DDoS. S každou ďalšou úlohou je firewall zaťažovaný. Platí to najmä v prípade ochrany proti DDoS. Vzhľadom na stavovú povahu firewallu je veľmi náchylný na útoky DDoS s vyčerpaním stavu. Odborníci na firewall neodporúčajú spoliehať sa pri obrane proti DDoS iba na firewall. Napriek tomu sa firewall dá použiť na ochranu proti DDoS. Úlohou firewallu je povoliť alebo zamietnuť prevádzku do privátnej siete na základe pravidiel. Spôsobov a pravidiel akými firewall môže chrániť sieť je viac, jedným z nich je filtrovanie paketov pomocou takzvaných ACL (Access Control List) listov, alebo napríklad aj použitím takzvaného WAF (Web Application Firewall). Filtrovanie paketov firewallom pomocou Access Control Lists bol jeden z prvých spôsobov ochrany pomocou firewallu. Na takéto filtrovanie sa zvyčajne používa smerovač nakonfigurovaný ako brána. Kontroluje sa zdrojová a cieľová IP adresa paketu, ale napríklad aj typ paketu a číslo portu, a na základe pravidiel v nakonfigurovanom ACL liste je paketu na smerovači buď povolený alebo zamietnutý prístup do siete. Pri tomto spôsobe filtrovania môže však nastať rovnaký problém ako pri už spomínanom filtrovaní na základe takzvaného blacklistu IP adries, a tým je falšovanie zdrojových IP adries útočníkmi, ktoré bude popísané v časti 2.4. Tým pádom môže byť obídená ochrana pomocou firewallu a ACL listov. Ďalším spôsobom ochrany pomocou firewallu je napríklad ochrana webovej aplikácie filtrovaním a monitorovaním HTTP prevádzky medzi aplikáciou a internetom, na to slúži takzvaný WAF (Web Application Firewall). Slúži na obranu siedmej vrstvy modelu OSI, vytvára akýsi štít pred webovou službou a škodlivej prevádzke zamedzuje prístup k nej. Výhodou WAF je rýchlosť a jednoduchosť akou sa dajú modifikovať jeho pravidlá, čo umožňuje pohotovú reakciu na meniace sa vektory útokov. Napríklad pri DDoS útoku je možné rýchlo modifikovať WAF pravidlá a zaviesť obmedzenie, takzvaný rate limiting. Táto metóda sleduje a obmedzuje požiadavky z rovnakej IP adresy ak prichádzajú príliš často, no mohla by byť útočníkmi obídená falšovaním IP adries.

#### Blokovanie pomocou vzoru v pakete

Ďalším zo spôsobov mitigácie je blokovanie pomocou určitého vzoru. Táto metóda mitigácie obmedzuje pakety útoku na základe ich obsahu. Na účely mitigácie je hľadaný určitý fixný reťazec v obsahu paketu a následne na základe neho prebieha blokovanie paketov v ktorých sa nachádza tento reťazec, sekvencia znakov, ktorý definuje útok. Okrem neho sa rovnako v praxi využíva aj vyhľadávanie a blokovanie pomocou určitého regulárneho výrazu. Tieto spôsoby sú používané komerčne, príkladom môže byť produkt Arbor od firmy Netscout, napríklad v Arbor Threat Mitigation System [1], implementuje mitigáciu aj týmto spôsobom. Avšak v žiadnych komerčných materiáloch nie sú uvádzané prístupy akým daný reťazec alebo regulárny výraz odvodzovať a blokovať na základe neho, preto je to predmetom skúmania tejto práce.

## 2.4 Falšovanie IP adries

Faktom je, že útočníci pri útokoch používajú rôzne techniky, ktoré sa neustále zlepšujú, aby bol útok silnejší, spôsobil väčšie škody a podobne. Jednou z používaných techník je aj skrývanie svojej identity, alebo vydávanie sa za niekoho iného. Na tento účel sa často používa takzvaný IP spoofing, útočník falšuje zdrojovú IP adresu za účelom vydávania sa za niekoho iného. Využívaný je aj pri DDoS útokoch aby sa znemožnilo ľahké spätné vystopovanie. Táto práca sa zaoberá odvodzovaním vzoru z payloadu paketu, ktorý môže byť neskôr použitý pre filtrovanie. Ako alternatíva, čo môže byť užitočné práve v prípadoch, kedy nie je až tak efektívne filtrovať útočiacu prevádzku na základe IP adries, ktoré môžu byť sfalšované. V nasledujúcej časti bude vysvetlený princíp falšovania IP adries podľa [12, 11, 21].

Falšovanie IP adresy, známe ako IP spoofing, má za cieľ skryť identitu odosielateľa alebo sa vydávať za iný počítačový systém a zabrániť tak spätnému vystopovaniu. Útočník vytvára a posiela pakety, v ktorých zámerne upraví IP hlavičku paketu tak, že v nej zmení údaj so zdrojovou IP adresou. Sfalšovanie zdrojovej adresy je často realizované náhodnými číslami, avšak adresa môže vyzerať ako adresa iného stroja. Ak je zdrojová IP adresa sfalšovaná a neustále náhodne menená, blokovanie škodlivých požiadaviek sa stáva zložitým.

Ak sa jedná o využitie falšovania IP adries pre realizovanie DDoS, tak sa táto metóda používa napríklad na nasledujúce účely:

#### Maskovanie zariadení botnetu a ich lokácie

V časti 2.2.1 bolo vysvetlené využitie botnetu pre účely DDoS útokov. Útočníci sa snažia maskovať zariadenia využívané v botnete pomocou falšovania zdrojových IP adries z viacerých dôvodov. Nechcú aby sa tieto zariadenia dostali na zoznamy útočiacich IP adries, ktoré sú tvorené ochrannými zariadeniami alebo službami, ktoré sa pokúšajú zmierniť útoky. To by mohlo spôsobiť, že zariadenia z botnetu by nemohli figurovať v ďalšom útoku lebo by boli filtrované na základe zoznamu. Ďalším dôvodom je že sa falšovaním snažia vyhnúť odhaleniu či už orgánmi činnými v trestnom konaní alebo v neposlednom rade aj odhaleniu zo strany majiteľov infikovaných zariadení, ktorí nevedia že ich zariadenie sa podieľa na nelegálnych útokoch.

#### Využitie pri reflexívnom DDoS útoku

Falšovanie zdrojovej IP adresy sa používa aj pri generovaní falošných požiadaviek na sprostredkovateľské servery, napríklad DNS alebo NTP servery. Tieto požiadavky vyzerajú ako

by ich vytvoril cieľ útoku, keďže zdrojová IP adresa sa často falšuje za adresu cieľa. Tým pádom odpovede na požiadavky prichádzajú zo serveru na cieľ, ktorý je nimi zahltený. Využíva sa tu často amplifikácia, z malých požiadaviek väčšinou vznikajú zložité veľké odpovede. Tento spôsob falšovania sa používa pri DNS amplifikácií alebo NTP amplifikácií.
Ďalšie využitie je pri takzvanom útoku smurf, [15]. Je to typ DDoS útoku, pri ktorom sú využívané ICMP Echo request pakety so sfalšovanou zdrojovou IP adresou posielané na broadcastovú IP adresu. Väčšina zariadení v sieti odpovedá na tieto pakety, ICMP Echo reply paketmi, a tým pádom dochádza k záplave obete, ktorej IP adresa bola použitá ako zdrojová. Veľkosť amplifikácie útoku zaleží od počtu zariadení v sieti.

## 2.5 N-gramy

Cieľom tejto práce je navrhnúť algoritmus pre odvodenie vzoru v DDoS útoku. Pri skúmaní tohto problému a návrhu ako k nemu pristupovať budú využité postupnosti, takzvané n-gramy. V tejto časti bude vysvetlený, definovaný, všeobecný význam a princíp n-gramov, podľa [4, 16, 7]. Samotný návrh pre riešenie cieľa tejto práce pomocou n-gramov bude popísaný v časti 3.2.

Pod pojmom n-gram, niekedy sa používajú aj názvy q-gram alebo k-mer, rozumieme súvislú postupnosť n položiek z danej vzorky, ktorou môže byť napríklad text, reťazec, reč alebo iné dáta. Položkami môžu byť napríklad slová, písmená, bajty podľa toho na aký účel sa n-gramy používajú. V prípade tejto práce budú vzorkami payloady paketov a položkami jednotlivé bajty v payloade, takže práca využíva takzvané byte n-grams, ktorými budú vlastne n-gramy tvorené z bajtov. Dĺžka n-gramu, čiže počet položiek, sa určuje podľa čísla n, ktoré patrí medzi prirodzené čísla. Na predpony v názvoch jednotlivých n-gramov sa používa buď latinský jazyk alebo anglický jazyk. V rámci tejto práce budú použité názvy podľa veľkosti nasledovne, n-gram s veľkosťou 1 sa nazýva unigram, s veľkosťou 2 bigram, s veľkosťou 3 trigram. Pre veľkosti 4, 5, 6 zasa nasledovne four-gram, five-gram, six-gram, a tak ďalej pre nasledujúce čísla.

N-gramy sa využívajú napríklad pri pravdepodobnosti, v počítačovej lingvistike, NLP, analýze biologických sekvencií, DNA sekvencií alebo pri kompresií dát. Časté využitie je pri N-gram modeloch na určenie pravdepodobnosti nasledujúceho slova. N-gramy sa môžu využívať aj na efektívne párovanie a porovnávanie. Aj v prípade tejto práce sú n-gramy, konkrétne n-gramy bajtov (byte n-grams), využité na reprezentáciu výskytov v payloadoch, výskyty sú spočítané, a následné získané štatistiky, z legitímnej a útočiacej prevádzky, sú porovnávané. Štatistiky obsahujú všetky skúmané n-gramy medzi ktorými sú aplikované porovnávacie pravidlá. Porovnávaním n-gramov v získaných štatistikách sa dajú detegovať anomálie, a neskôr podľa nich odvodiť vzor, ako bolo spomenuté vyššie, návrh bude popísaný v časti 3.2.

Príklad tvorby n-gramov, kde položkou je znak, je znázornený na nasledujúcej ukážke. Pri tvorbe n-gramov je dôležitý výber všetkých kombinácii a dochádza zároveň k prekrývaniu jednotlivých položiek medzi n-gramami, keďže pri tvorbe n-gramu je posúvané, o jednu pozíciu vo vzorke, posúvacie okno s dĺžkou n. Rovnakým princípom je inšpirovaná tvorba návrhu v časti 3.2. Pre názornú ukážku bolo vybrané slovo `mitigácia`. Znázornené sú unigramy až trigramy. Jednotlivé n-gramy sú teda nasledujúce:

- unigramy: m, i, t, i, g, á, c, i, a
- bigramy: mi, it, ti, ig, gá, ác, ci, ia
- trigramy: mit, iti, tig, igá, gác, áci, cia

# Kapitola 3

# Návrh

V tejto kapitole práce bude popísaný problém, ktorým sa zaoberá táto práca, ďalej budú popísané navrhované a skúmané možné metódy riešenia problému. Pri metódach bude odkazované na teoretické časti, ktoré sú dôležitým podkladom pre návrh a skúmanie metód, a popísané boli v kapitole 2. Z navrhovaných metód bude vybraná jedna, konkrétne metóda s využitím n-gramov, ktoré boli vysvetlené v 2.5. Implementácia tejto metódy bude popísaná v kapitole 4 a následné experimenty na dátových sadách a vyhodnotenie metódy bude popísané v kapitole 5.

Problematika, ktorá je podnetom pre túto prácu je neustála snaha útočníkov obísť alebo zamedziť blokovanie ich útokov a na druhej strane obrancov zabezpečiť dostatočnú obranu dostatočným blokovaním útokov. V mnohých prípadoch útočníci využívajú spôsob falšovania zdrojovej IP adresy, ktorý bol popísaný v časti 2.4. Obrancovia sa v takomto prípade nemôžu spoliehať na blokovanie na základe IP adries a preto hľadajú iné spôsoby, ktoré by sa dali použiť na identifikáciu útočných paketov. Predpoklad, na ktorom sa zakladá aj motivácia tejto práce je, že útočníci zanechávajú v paketoch útoku nejaký vzor, buď neúmyselne alebo to vyžaduje typ útoku. Príkladmi na podporenie tohto predpokladu môžu byť napríklad typy útokov, ktoré majú za cieľ využiť určitú službu. Tak je potreba vygenerovať požiadavky, napríklad požiadavky na určité doménové meno alebo typ záznamu, spôsobom kedy je výsledkom čo najväčšia odpoveď. Týmito typmi útokov sú napríklad, DNS alebo NTP reflexívno-amplifikačné útoky, ktoré boli popísané v časti 2.2.2. Vzor v paketoch môže byť zanechaný v hlavičke paketu alebo v payloade paketu. Táto práca sa zaoberá odvodením vzoru z payloadu paketu. Na druhej strane metódou odvodenia vzoru z hlavičky paketu sa už zaoberá napríklad bakalárska práca [13].

Cieľom tejto práce je navrhnúť metódu pre odvodenie vzoru, sekvencie bajtov, z payloadu paketov, ktoré sú majoritne zastúpené v DDoS útokoch. Pojem payload bol vysvetlený v časti 2.1. Metóda by mala byť schopná odvodiť vzor v útočiacej prevádzke, ktorý čo najviac charakterizuje pakety patriace útoku. Odvodený vzor by mal slúžiť a pomôcť pri následnej mitigácií zmierniť prebiehajúci útok. Pri odvodení vzoru by si mala navrhnutá metóda zachovať čo najmenšiu mieru falošne pozitívnych paketov, čiže by nemala obmedziť legitímnu prevádzku, čo by napomohlo útočníkovi. Metóda bude navrhnutá spôsobom, že ide o reakciu na útok po predošlej detekcií útoku, zároveň predpokladá odovzdanie príslušných dát. Metóda teda útok nedeteguje ale je to reakcia na detegovaný útok s dodatočnou analýzou dát. Cieľom tejto práce zároveň ale nie je navrhnúť metódu, ktorá by prispela k blokovaniu všetkých typov DDoS útokov, ale k blokovaniu takých typov DDoS útokov, ktoré nejdú blokovať inak ako na základe odvodeného vzoru z payloadu. Je to teda jedna z mnohých metód blokovania DDoS útokov.

V rámci tejto práce bolo po dohode s vedúcim práce navrhnutých a skúmaných viacero metód pre danú problematiku. Metódy boli skúmané a zvažovala sa ich vhodnosť pre danú problematiku. Metódy, ich navrhované využitie a zhodnotenie budú popísané v nasledujúcej časti tejto kapitoly, konkrétne 3.1. Následne bola vybraná jedna metóda, konkrétne metóda pristupujúca k problému pomocou n-gramov, ktorá bola vybraný ako možno najvhodnejšia metóda z možných skúmaných metód. Návrh vybranej metódy využívajúcej n-gramy bude popísaný v časti 3.2.

## 3.1 Návrh a zhodnotenie možných prístupov

V tejto časti práce budú popísané a zhodnotené návrhy metód, ktoré boli zvažované pre problematiku odvodzovania vzoru z payloadu paketu, ktorým je motivovaná táto práca. Medzi metódami nie je popísaný návrh metódy, ktorá bola použitá a s ktorou sa robili experimenty, ktoré budú popísané v kapitole 5. Návrh vybranej metódy bude popísaný v časti 3.2.

V nasledujúcich častiach budú popísané zvažované metódy riešenia problematiky tejto práce:

### 3.1.1 Prístup pomocou strojového učenia

Medzi prístupmi, ktoré boli skúmané ako možné riešenia problému je aj využitie strojového učenia, konkrétne metódou použitia neurónových sietí. Neurónové siete majú vlastnosť abstrakcie pravidiel medzi vstupnými hodnotami a výstupnými hodnotami. Následne môžu byť získané pravidlá použité na akékoľvek vstupné hodnoty. Proces abstrakcie sa nazýva učenie. Po ukončení učenia, sieť produkuje výstupy podľa uvedeného pravidla aplikovaného na vstupné hodnoty. Učenie neurónových sietí by sa dalo využiť v rámci tejto práce spôsobom, že by vstupom neurónovej siete boli payloady paketov legitímnej prevádzky. Neurónová sieť by sa teda učila na rôznych dátach, v ktorých by sa nachádzala iba legitímna prevádzka. Toto učenie môže prebiehať v časoch keď neprebieha DDoS útok, čiže na učenie neurónovej siete by bol dostatok času. Predpokladom je, že následne by počas útoku naučená neurónová sieť bola schopná s určitou pravdepodobnosťou rozpoznávať a označovať podľa pravidiel payloady paketov patriace legitímnej prevádzke ale zároveň aj tie, ktoré patria útočiacej prevádzke.

Metóda použitia neurónových sietí sa teda javí, že by vedela odhaliť pakety, ktoré patria útočiacej prevádzke, na základe učenia sa na legitímnej prevádzke. Toto riešenie tejto metódy ale nerieši celkový cieľ tejto práce, ktorým je odvodenie vzoru, ktorý definuje daný DDoS útok, z payloadu paketu. Výstupom prístupu použitia tejto metódy je samotný model v podobe neurónovej siete, ktorý je naučený na legitímnej prevádzke, ktorý síce vie označiť pakety nepatriace legitímnej prevádzke ale nevie odvodiť konkrétny vzor z payloadu paketov. Tento model sa navyše ale ani pre vyhodnocovanie nad paketmi v súčasných sieťových zariadeniach, napríklad smerovačoch alebo IDS, nejaví ako vhodný, čo je ďalšou nevýhodou tejto metódy.

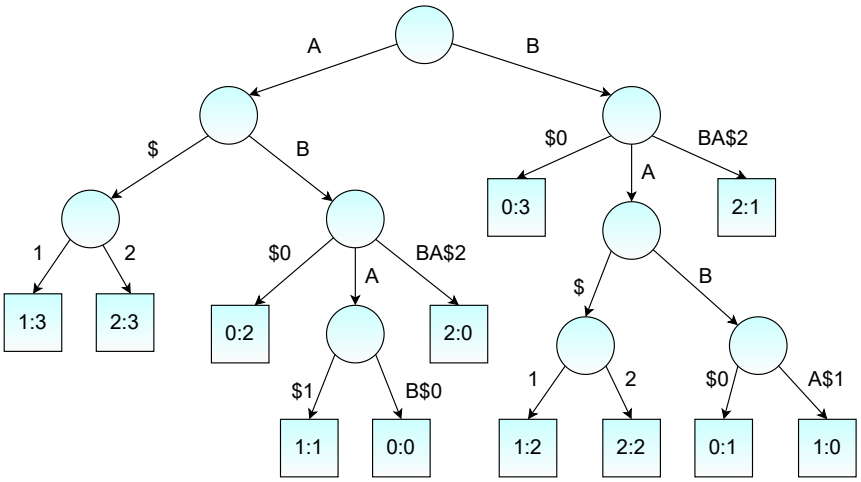
Využitie neurónovej siete sa teda momentálne nejaví ako vhodné riešenie pre odvodenie samotného vzoru z payloadu pre mitigáciu DDoS útoku, čo je cieľom tejto práce. Preto sa nerobili experimenty pomocou tohto prístupu a zámer tejto práce bol presunutý na iné metódy, ktoré boli skúmané.

### 3.1.2 Prístupy hľadaním najdlhšieho spoločného podreťazca

Ďalšie možné navrhované riešenia, problematiky ktorú rieši táto práca, boli inšpirované metódami, ktorými sa rieši problém najdlhšieho spoločného podreťazca, čiže takzvaný Longest Common Substring problem. Tento problém spočíva v hladaní najdlhšieho možného reťazca, ktorý je zároveň podreťazcom každého reťazca v ktorom je daný spoločný podreťazec hladaný. Metóda môže vyhľadávať najdlhší spoločný podreťazec medzi dvoma ale aj viac reťazcami. Problém môže mať viacero riešení. Longest Common Substring problem sa rieši napríklad metódou pomocou dynamického programovania alebo využitím takzvaných Suffix Trees. Obidve tieto metódy boli zvažované na využitie v tejto práci, ich princíp použitia a vyhodnotenie bude popísné v nasledujúcich častiach, použité informácie boli inšpirované z [3, 8, 5].

#### Prístup pomocou Suffix tree

Problém najdlhšieho spoločného reťazca sa môže riešiť pomocou vytvorenia takzvaného zovšeobecneného suffix tree. Zovšeobecnený suffix tree sa dá definovať ako zovšeobecnený strom skladajúci sa z prípon všetkých daných reťazcov, v ktorých sa hľadá najdlhší spoločný reťazec. Suffix tree obsahuje presne $n$ listov, kde $n$ je celkový počet znakov daných reťazcov. Každý vnútorný uzol má najmenej dvoch potomkov. Každá hrana je označená neprázdnym podreťazcom, a žiadne dve hrany vychádzajúce z rovnakého uzla nemôžu byť označené podreťazcom začínajúcim rovnakým znakom. Pre názornú ukážku bol zostavený zovšeobecnený suffix tree, pre reťazce $ABBA$, $ABAB$, $BABA$, označené ako reťazce 0,1 a 2, znázornený na obrázku 3.1. Za každým reťazcom sa nachádza ukončovací znak a číselné označenie reťazca.




Obr. 3.1: Suffix tree pre reťazce $ABBA$, $ABAB$, $BABA$, inšpirované obrázkom z odkazu[1]

Po zostrojení suffix tree následné nájdenie najdlhšieho spoločného podreťazca daných reťazcov, prebieha nájdením najhlbších vnútorných uzlov, ktoré majú listové uzly, v podstrome pod ním, zo všetkých daných reťazcov. Na obrázku 3.1, sú v listoch číselne označené reťazce, v ktorých sa daný podreťazec nachádza, a za dvojbodkou index na ktorom sa daný podreťazec v reťazci nachádza. Najdlhším spoločným podreťazcom v tomto prípade sú reťazce dva, $AB$ a $BA$.

[1] https://commons.wikimedia.org/wiki/File:Suffix_tree_ABAB_BABA_ABBA.svg

Tento prístup sa teda javil, že by mohol byť vhodný aj v prípade problematiky ktorú rieši táto práca. Návrh využitia riešenia pomocou suffix tree, bol rovnako ako pri reťazcoch vytvoriť suffix tree z payloadov paketov, v ktorých má byť hľadaný vzor. Následne by sa hľadal vo vytvorenom suffix tree najdlhší spoločný reťazec, ktorý by sa dal vyhlásiť ako vzor nájdený v payloadoch paketov. Na základe dodatočných pravidiel a skúmania by bol získaný vzor vyhodnotený, či dobre definuje DDoS útok. No po skúmaní zložitosti tohto prístupu, hlavne priestorovej, a kvôli problému, ktorý bude spomenutý nižšie, bol zámer tejto práce presunutý na iné skúmané metódy. Zostavenie suffix tree zaberie $O(N)$ času, kde $N$ je súčet dĺžok všetkých reťazcov, v ktorých je podreťazec hľadaný. K časovej zložitosti treba ešte pripočítať časovú zložitosť potrebnú na vyhľadanie najdlhšieho spoločného reťazca, tá je daná ako $O(N \cdot K)$, kde $K$ je počet daných reťazcov, čiže payloadov paketov. Ďalším problémom, ktorý sa vyskytol pri skúmaní tohto prístupu a vyskytol sa aj pri prístupe pomocou dynamického programovania, bol prípad keď aspoň jeden z reťazcov, čiže payloadov paketov neobsahoval žiadny spoločný podreťazec, tento problém, aj zvažovaný prístup pomocou dynamického programovania bude popísaný nižšie, v 3.1.2.

#### Prístup pomocou dynamického programovania

Ďalším spôsobom, akým sa môže riešiť problém najdlhšieho spoločného reťazca je pomocou použitia dynamického programovania. Algoritmus, je vhodný na použitie pre hľadanie medzi dvoma reťazcami, no môže byť rozšírený na hľadanie medzi viacerými reťazcami. Podstata spočíva v hľadaní najdlhšieho spoločného podreťazca zo všetkých podreťazcov vo všetkých daných reťazcoch. Algoritmus si pri hľadaní najdlhšieho podreťazca ukladá doteraz nájdený spoločný podreťazec, a ak nájde dlhší tak uložený podreťazec aktualizuje. Pre hľadanie medzi dvoma reťazcami je všeobecná časová aj priestorová zložitosť $O(m \cdot n)$, kde $m$ a $n$ sú dĺžky reťazcov, priestorová zložitosť však môže byť znížená efektívnym ukladaním až na $O(\min(m, n))$. Pri hľadaní medzi viac ako dvoma reťazcami zaberá prístup pomocou dynamického programovania $O(n_1 \cdot \ldots \cdot n_K)$ času a $O(n_1 + \ldots + n_K)$ priestoru, kde $K$ je počet daných reťazcov.

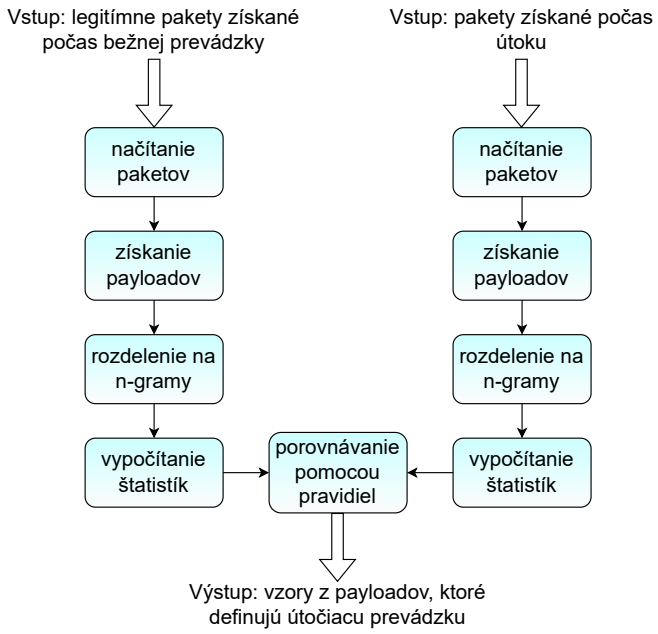
Prístup pomocou dynamického programovania sa tiež javil, že by mohol byť vhodný aj v prípade problematiky ktorú rieši táto práca. Návrh využitia riešenia bol využiť algoritmus dynamického programovania na vyhľadávanie v payloadoch paketov, rovnako ako vyhľadáva v reťazcoch. Hľadaný najdlhší spoločný podreťazec by vo výsledku znamenal a reprezentoval nájdenie spoločného vzoru, sekvencie bajtov, v payloadoch paketov. Na základe dodatočných pravidiel a skúmania by bol získaný vzor vyhodnotený, či dobre definuje DDoS útok. No po skúmaní použitia tohoto prístupu bol zistený nedostatok, ktorý tento prístup má, spoločne aj s prístupom pomocou suffix tree, popísaným v časti 3.1.2. Nedostatkom bol prípad keď aspoň jeden z reťazcov, čiže payloadov paketov neobsahoval žiadny spoločný podreťazec. Takže napríklad, keď by štyri z piatich reťazcov obsahovali spoločný podreťazec, a jeden reťazec neobsahoval ani jeden podreťazec spoločný s ostatnými, tak by algoritmus vyhodnotil za najdlhší spoločný podreťazec prázdny reťazec. To aj napriek tomu, že štyri reťazce obsahujú spoločný podreťazec. V problematike tejto práce sa takýto prípad vyskytuje bežne. Napríklad keď by spomínané štyri reťazce boli payloady paketov, ktoré patria DDoS útoku, takže predpokladom je že obsahujú spoločný vzor, a piaty reťazec by bol payload paketu legitímnej prevádzky, ktorý sa odlišuje od útočiacej prevádzky, tak by dané prístupy neodvodili vzor útoku a vyhodnotili vzor ako prázdny reťazec. Tento nedostatok, môže byť predmetom skúmania v prípadnom nadviazaní na túto prácu. V tejto práci sa zámer presunul na iný prístup, ktorý bol zvolený pre následnú implementáciu a experi-

menty, to konkrétne prístup k problému pomocou n-gramov, ktoré boli popísané v časti 2.5. Zvolený prístup pomocou n-gramov bude popísaný v nasledujúcej časti 3.2.

## 3.2 Zvolený prístup pomocou využitia n-gramov

Pre riešenie odvodenia vzoru z payloadu paketov pre mitigáciu DDoS útoku bol zvolený na implementáciu a následné experimenty prístup pomocou použitia takzvaných n-gramov, ktoré boli popísané v časti 2.5. V tejto časti bude popísaný navrhnutý pracovný postup, ktorý bol použitý a s ktorým sa vykonávali experimenty nad payloadami paketov.

Prístup k problému s využitím n-gramov sa javil ako najvhodnejší zo skúmaných prístupov. Rozhodnutie pre jeho využitie a experimenty či je naozaj vhodný na problematiku tejto práce, bolo inšpirované využívaním n-gramov v problematike pri analýze biologických sekvencií a DNA sekvencií. Jedným z dôvodov výhody tohto prístupu je, že v tomto prípade výstupom sú vzory, sekvencie bajtov, ktoré môžu definovať útok aj napriek tomu, že sa nemusia nachádzať v každom skúmanom payloade paketu. Čo je teda predpokladom, pretože v dátovej sade na vstupe tohto prístupu, medzi útočiacimi paketmi ktoré sú definovateľné nejakým vzorom v payloade a tých bude väčšina, budú aj pakety, ktoré tento vzor v payloade nemajú. Teda tento prístup nezlyháva na probléme, ktorý sa vyskytol pri prístupoch hľadaním najdlhšieho spoločného podreťazca. V prípade tohoto prístupu sú z payloadu paketov zostavované dané n-gramy a zisťované počty výskytov. Na základe porovnávania s výskytmi n-gramov získanými z payloadov paketov v čase o ktorom vieme povedať, že v ňom neprebiehal DDoS útok a nachádzala sa v ňom iba bežná legitímna prevádzka, sú vyhodnotené výsledné n-gramy, ktoré sa dajú považovať ako vzory. Na obrázku 3.2 je priblížený navrhovaný pracovný postup metódy prístupom pomocou n-gramov.




Obr. 3.2: Navrhnutý postup odvodenia vzoru z payloadu s využitím n-gramov

Metóda predpokladá predanie príslušných dát na jej vstup. Vstupom sú predané dáta, ktoré obsahujú pakety prevádzky. Metóda počíta nad získanými dátami štatistiky a následne ich porovnáva pravidlami s inou štatistikou. Ako je vidno na obrázku 3.2, metóda spočíva

v porovnávaní získaných štatistík z dvoch časov. Jedna je vypočítaná z predaných dát z času, keď sa dá povedať, že v sieti neprebiehal žiadny útok, takže pakety patria legitímnej prevádzke. Ďalšia štatistika je vypočítaná z predaných dát a času, keď v útoku prebieha útok a z týchto dát má byť odvodený vzor. Dáta s legitímnou prevádzkou môžu byť rôzne veľké, záleží akú dlhú dobu boli získavané. Môžu byť tak pomerne o dosť väčšie ako dáta z čias útoku. Experimenty s touto metódou budú vykonávané, takzvane offline, a použité dátové sady budú popísané v časti 5.1. Medzi týmito sadami sa teda nachádzajú aj dáta s legitímnou prevádzkou a aj dáta s útočiacou prevádzkou.

Treba však spomenúť, že metóda sa zakladá na predpoklade, že útočiace pakety budú sebe podobné. Podobnosť sa predpokladá v rámci obsahu payloadu paketov. Ďalším z predpokladov je, že pakety patriace DDoS útoku budú zastúpené vo väčšine v dátach ktoré boli získané počas útoku.

Návrh vypočítania samotných štatistík by sa dal rozdeliť do štyroch krokov, znázornené aj na obrázku 3.2. Tento postup sa vykonáva samostatne pre unigramy, bigramy a trigramy. Kroky sú načítanie paketov, získanie payloadov, rozdelenie payloadov na konkrétne n-gramy a samotné vypočítanie štatistík. Po vykonaní týchto krokov a vypočítaní štatistík sa štatistiky medzi sebou porovnajú navrhnutými pravidlami.

Načítavanie paketov prebieha postupne načítaním každého paketu z dát ktoré boli metóde predané. Metóda následne získa z paketu payload, tým že načíta dáta ktoré obsahuje UDP alebo TCP paket, čiže transportná vrstva. Pre lepšie spracovanie a rozdelenie do n-gramov sú bajty ktoré payload obsahuje vyjadrené v hexadecimálnej reprezentácií. Následne je v ďalšom kroku payload rozdelený na menšie časti, sekvencie bajtov, ktorými sú n-gramy, popísané boli v časti 2.5. V tomto prípade je vzorkou payload paketu a položkami sú jednotlivé bajty payloadu. Pre ukladanie štatistík n-gramov budú použité matice s pevným počtom riadkov, aby sa riadky dali efektívne indexovať pomocou vyššie spomínanej hexadecimálnej reprezentácií. Pre tento prístup bude skúmané využitie od unigramov až po trigramy. Zvolené veľkosti boli vybrané ako prvotný prístup pre skúmanie tejto metódy, zároveň väčšie n-gramy neboli vybrané aj kvôli pamäťovej náročnosti. Ukladanie všetkých možných kombinácií four-gramov a väčších n-gramov by bolo už priestorovo náročné. Skúmanie využitia a prípadné rozšírenie metódy na dlhšie bajtové sekvencie môže byť predmetom skúmania v rozšírení tejto práce do budúcna alebo nadviazanie na ňu.

V ďalšom kroku sú nad jednotlivými n-gramami počítané štatistiky, pre každý n-gram sú ukladané dve hodnoty. Týmito hodnotami sú počet výskytov daného n-gramu vo všetkých paketoch z dát ktoré boli metóde predané a počet paketov, v ktorých sa daný n-gram vyskytoval. Tieto hodnoty sú aktualizované každým spracovaným paketom. Ak sú spracované všetky pakety tak štatistika je pripravená na porovnávanie pomocou pravidiel.

Porovnávanie získaných štatistík n-gramov z payloadov paketov prebieha pomocou dvoch pravidiel. Pravidlá detegujú anomálie porovnávaním štatistík, každé pravidlo sa pritom zaoberá jednou z dvoch ukladaných hodnôt v štatistikách. Jedno z pravidiel sa zameriava na počet výskytov daného n-gramu vo všetkých paketoch z dát ktoré boli metóde predané a druhé pravidlo na počet paketov, v ktorých sa daný n-gram vyskytoval. Pre každé pravidlo je nastavovaný určitý prah, takzvaný threshold. Správne nastavenie thresholdov spolu s overením použiteľnosti metódy na problematiku tejto práce bude predmetom experimentov, ktoré budú popísané v kapitole 5. Pravidlá boli navrhnuté na základe anomálií, ktoré sa môžu vyskytovať v paketoch pri útoku. Pravidlá na základe nastaveného thresholdu vyhodnotia určité n-gramy za podozrivé. Pravidlá budú popísané nižšie.

Za anomálie v sade s útokom oproti tej s legitímnou prevádzkou sa môžu považovať nasledujúce predpoklady. Počet výskytov nejakého n-gramu je veľký v štatistike z dát s útokom

a v štatistike z dát s legitímnou prevádzkou je oproti tomu tento počet pre daný n-gram značne nižší alebo nulový. Tak sa daný n-gram považuje za podozrivý, pretože môže indikovať vzor. Tento predpoklad sa zakladá na vyššie spomínanom predpoklade, že pakety útoku sú zastúpené vo väčšine v útočiacej dátovej sade a ich payloady sú sebe podobné. Ďalším predpokladom kedy n-gram môže byť vzor je napríklad aj prípad, kedy celkový počet výskytov daného n-gramu je podobný počtu paketov v ktorom sa daný n-gram vyskytuje. N-gram v tom prípade môže indikovať vzor, sekvenciu bajtov, ktorý je približne jedenkrát v každom útočiacom pakete. Môže to teda reprezentovať napríklad určitú rovnakú odpoveď od DNS alebo NTP servera pri reflexívno-amplifikačnom útoku. Tento predpoklad by bol však viac relevantný pre dlhšie n-gramy ako trigramy, no táto práca ako prvotný prístup skúmania na overenie použiteľnosti metódy sa zameriava na menšie n-gramy.

Na základe predpokladov sa vytvorili pravidlá, ktoré sú aplikované a použité na porovnávanie získaných štatistík. Ako bolo vyššie spomenuté pravidlá sú dve. Pri tvorení pravidiel bolo cieľom vytvoriť pravidlá, ktoré by boli čo najmenej, respektíve vôbec obmedzené na určitú veľkosť dátovej sady, čiže sa dbalo na všeobecnosť. Spomínané hodnoty zo štatistík boli normalizované a bol vytvorený pomer v ktorom hodnoty boli predelené počtom unikátnych n-gramov v danej dátovej sade. Prvé pravidlo je formálne zapísané nasledovne,

$$\frac{PV_N}{PU_N} - \frac{PV_L}{PU_L} >= T_1$$

a druhé pravidlo zasa nasledovne,

$$\frac{PP_N}{PU_N} - \frac{PP_L}{PU_L} >= T_2$$

kde hodnoty $T_1$ a $T_2$ v pravidlách značia thresholdy pre konkrétne pravidlo, $PV$ je počet výskytov daného n-gramu, $PP$ je počet paketov v ktorých sa daný n-gram nachádzal a $PU$ je počet unikátnych n-gramov v dátovej sade. Indexy $L$ a $N$ naznačujú, či sa jedná o dátovú sadu s legitímnou alebo nelegitímnou prevádzkou.

Následne sa z množín obsahujúcich n-gramy, ktoré označili jednotlivé pravidlá za podozrivé, spraví zjednotenie a výsledná množina obsahuje vzory. Výstupom navrhnutej metódy sú teda skupiny unigramov, bigramov a trigramov, ktoré boli vyhodnotené pravidlami ako podozrivé a ktoré sa tým pádom považujú za vzory. Tieto odvodené vzory sú považované za definujúce útok a pakety, v ktorých sa nachádza odvodený vzor sú považované za pakety patriace útoku. Metóda teda však sama o sebe neoznačuje pakety patriace útoku ale odvodzuje vzory a tieto vzory môžu byť neskôr použité na mitigáciu. Pre testovanie a vyhodnotenie metódy bolo však potrebné označiť pakety. Pre toto označenie bolo treba určiť určitý prah, spodnú hranicu, pri ktorom sa paket považoval za paket útoku. Určovanie prebieha zistením koľko percent paketu tvoria n-gramy ktoré boli na výstupe metódy. Keďže v primárnych experimentoch nešlo o zistenie vhodnej hodnoty pre tento prah, tak bol nastavený na nižšiu hodnotu a to 20%.

Horná teoretická hranica priestorovej náročnosti tejto metódy pre určitý n-gram je $O(256^n)$ pre uloženie všetkých možných kombinácií daného n-gramu. Konštanta 256 reprezentuje počet všetkých možných znakov na $n$, čiže počet možných znakov ktoré sa v bajte paketu môže nachádzať. Pre unigramy je to $256^1 = 256$, pre bigramy $256^2 = 65536$ a pre trigramy $256^3 = 16777216$. Horná teoretická časová náročnosť je pre spracovanie konkrétneho payloadu (zistenie všetkých nachádzajúcich sa n-gramov konkrétnej dĺžky) lineárna, čiže $O(n)$, kde $n$ je dĺžka payloadu v bajtoch. Následne pri porovnávaní pravidiel je časová náročnosť taktiež lineárna, $O(n)$, kde $n$ je počet riadkov matice a každý riadok matice odpovedá štatistike o jednom určitom n-grame.

# Kapitola 4

# Implementácia

V tejto kapitole bude popísaná implementácia jednotlivých častí a experimentov zvoleného riešenia, ktorého návrh bol popísaný v časti 3.2. Popísané budú aj použité nástroje.

## 4.1 Vyvojové prostredie

Programová časť tejto práce bola implementovaná použitím jazyka Python, konkrétne s verziou 3.8.6. Na účely experimentov bol zároveň využitý Jupyter Notebook, pre jeho vhodnosť na experimentovanie. Pre vizualizáciu experimentov pomocou grafov bola využitá knižnica Matplotlib[1]. Na namiešanie dátových sád, manuálnu kontrolu a overovanie výstupov programu bol použitý nástroj Wireshark[2]. Pre účely zálohovania bol použitý distribuovaný verzovací systém Git[3]. Hlavný program, v ktorom je implementácia navrhnutej metódy je v súbore `mitigation-ddos.py`.

## 4.2 Vstup programu

S navrhnutou metódou, ktorá bola popísaná v časti 3.2, boli vykonávané takzvané offline experimenty. Vstupom preto sú dátové sady, popísané budú v časti 5.1, uložené v `.pcap` súboroch. Jedným z dvoch vstupných súborov je dátová sada obsahujúca legitímnu prevádzku, vstupný súbor sa zadá parametrom `--legit`. Druhým súborom je dátová sada, ktorá obsahuje prevádzku s paketmi útoku, tento vstupný súbor sa zadá parametrom `--ddos`.

Ďalšími vstupnými hodnotami programu, predávanými dvoma parametrami `--threshold1` a `--threshold2` sú číselné hodnoty ktorými sú nastavované jednotlivé prahy pravidiel. V prvom pravidle je nastavovaný threshold pomocou parametru `--threshold1`. Threshold, ktorý je nastavovaný v druhom pravidle je zasa predávaný parametrom `--threshold2`. Navrhnuté pravidlá boli popísané v časti 3.2. Tieto dve číselné hodnoty sú typu `float`.

Vstupným parametrom je aj súbor do ktorého má byť uložený výsledok metódy, teda nájdené vzory. Tento súbor je predávaný programu parametrom `--output-file`, odporúčaný je súbor `.txt`.

Ďalšie vstupy programu sú nepovinné. Prvým je dátová sada, uložená v `.pcap` súbore, na overenie výsledkov, takzvaná testovacia sada. Táto testovacia sada by mala obsahovať rovnaký typ útoku na akom bola metóda natrénovaná, no pakety by mali byť odlišné. Táto

[1] https://matplotlib.org/
[2] https://www.wireshark.org/
[3] https://git-scm.com/

sada je predávaná parametrom `--eval-file`. Spolu s týmto vstupom musí byť zadaný aj rozsah čísiel paketov, ktoré značia pakety, ktoré v testovacej sade patria útoku. Toto rozmedzie sa zadáva pomocou parametru `--eval-range`. Ak sú tieto dva spomínané parametre zadané, tak okrem štandardnej funkcionality tejto metódy sa vykoná aj evaluácia. Evaluácia spočíva v overení do akej miery výstupné vzory definujú útočiacu prevádzku. Výstupom sú percentuálne hodnoty značiace mieru skutočne pozitívne označených paketov a mieru falošne označených paketov. Táto evaluácia bola využitá na vykonané experimenty, ktoré budú popísané v časti 5.2.

## 4.3 Funkcionalita a použité knižnice

V tejto časti bude popísaná implementácia jednotlivých krokov, ktoré boli znázornené na obrázku 3.2 a popísané v časti 3.2. Jednotlivé kroky boli implementované a rozdelené do procedúr.

Načítavanie paketov prebieha zo vstupných súborov s dátovými sadami vo formáte `.pcap`. Načítané a spracované sú najskôr pakety zo súboru s legitímnou prevádzkou a následne zo súboru s útočiacou prevádzkou. Na načítavanie paketov bola využitá knižnica `Scapy`, konkrétne funkcia `sniff`. `Scapy` je nástroj slúžiaci na manipuláciu s paketmi. Je to program v jazyku Python, ktorý môže byť použitý napríklad na dekódovanie, falšovanie, zachytávanie, posielanie paketov a tak ďalej. Pakety sú načítavané z dátovej sady po jednom a sú hneď spracovávané, po spracovaní sa neukladajú do pamäte.

Z načítaného paketu je v ďalšom kroku získaný jeho payload, konkrétne pomocou ukazovateľa `payload` z triedy `Packet` knižnice `Scapy`. Ukazovateľ ukazuje na dáta paketu zvolenej vrstvy, v prípade tejto práce sú to dáta, ktoré obsahuje transportná vrstva, UDP alebo TCP pakety. Pre lepšie spracovanie a rozdelenie do n-gramov je payload z paketu ukladaný v jeho hexadecimálnom výpise, teda každý bajt payloadu je reprezentovaný ako dvojmiestne hexadecimálne číslo.

V ďalšom kroku je payload rozdelený na n-gramy, toto prebieha samostatne pre unigramy, bigramy a trigramy. Všeobecný príklad tvorby n-gramov bol znázornený na príklade v časti 2.5. Každý n-gram z payloadu je teda postupnosť bajtov s dĺžkou $n$. Na ukladanie celkových štatistík pre každý n-gram boli zvolené matice reprezentované 2d poliami, pomocou knižnice `numpy`. Globálne matice sú celkovo tri, konkrétne pre unigramy, bigramy a trigramy, každá s rozmermi $256^n \times 2$. Každý riadok v matici reprezentuje štatistiku o určitom n-grame, v prvom stĺpci je číslo reprezentujúce celkový počet výskytov daného n-gramu v payloadoch paketov a v druhom stĺpci je číslo reprezentujúce počet paketov v ktorých sa daný n-gram vyskytoval. Pri spracovaní konkrétneho payloadu paketu je uchovávaná informácia o počtoch n-gramov v slovníku, konkrétne `defaultdict` z knižnice `collections`. Následne po spracovaní celého paketu sú na základe tohto slovníka aktualizované hodnoty vo vyššie spomínaných globálnych maticiach pre každý n-gram. Keďže sú jednotlivé bajty payloadu reprezentované ako dvojmiestne hexadecimálne číslo, tak je n-gram sekvencia týchto čísel a dá sa pomocou samotných n-gramov jednoducho indexovať riadok v matici. Na konkrétnom riadku sú následne navyšované hodnoty. Ak sa spracujú všetky pakety tak sú štatistiky, uložené v týchto troch maticiach, nachystané pre porovnávanie pomocou pravidiel.

Porovnávanie pomocou pravidiel je implementované presne na základe formálnych zápisov pravidiel, ktoré boli popísané v časti 3.2. Porovnávajú sa dve štatistiky, legitímna a nelegitímna, pre každý n-gram. Použité sú na to hodnoty štatistiky ktoré boli uložené vo vyššie spomínaných maticiach. V každom pravidle je použitý určitý threshold. Thres-

holdy pre jednotlivé pravidlá boli predané metóde na vstupe. Ak aspoň jedno pravidlo označí konkrétny n-gram, čiže výsledok porovnávania s thresholdom bude hodnota `True`, tak je n-gram pridaný do poľa označených n-gramov ako vzory útoku. To znamená že medzi pravidlami sa spraví zjednotenie, čiže logický `or`.

Ak sú zadané parametre na evaluáciu výsledných vzorov, tak sa zo zadanej testovacej dátovej sady načítavajú pakety pomocou funkcie `sniff`, z knižnice `Scapy`. V každom spracovávanom pakete sú vyhľadávané výsledné vzory, ak aspoň 20% paketu tvoria výsledné vzory tak je označený za pozitívny. Po spracovaní všetkých paketov sú spočítané metriky, pakety označené ako pozitívne sa porovnávajú či naozaj patria do rozmedzia pozitívnych. Toto rozmedzie je zadané spomínaným parametrom `--eval-range`. Výsledné metriky sú vypísané a to konkrétne percentuálne hodnoty skutočne pozitívnych paketov a falošne pozitívnych paketov.

## 4.4 Výstup programu

Výstupom programu sú vzory, n-gramy ktoré definujú útok. Tieto vzory sú zapísané do výstupného súboru. Pretože vzory odvodené z payloadu paketu môžu obsahovať aj netlačiteľné znaky tak sú tieto vzory zapísané do súboru v ich hexadecimálnej reprezentácií. Vzory sú zapísané v súbore nasledovne, na prvom riadku sú zapísané unigramy oddelené čiarkou, na druhom riadku bigramy oddelené čiarkou a na treťom riadku trigramy oddelené čiarkou.

# Kapitola 5

# Testovanie

## 5.1 Dátové sady

Pre vyhodnotenie a experimenty s navrhnutou metódou bolo potrebné nachystať rôzne dátové sady. Pre účely tejto práce boli prevzaté dátové sady z práce [13], ktoré boli doporučené vedúcim tejto práce. Na túto prácu boli použité sady:

- **LEGIT** - ako sa v spomínanej práci píše, prevádzka v tejto dátovej sade bola zachytená na chrbtovej sieti medzi sieťami ACONET a CESNET, a obsahuje približne 76800 paketov, medzi ktorými sa nachádza 40% TCP paketov. Z tejto dátovej sady boli na účel tejto práce vyberané rôzne počty náhodných paketov a boli namiešavané s paketmi útoku. Počty budú spomenuté pri jednotlivých experimentoch.
- Dátová sada nazvaná **LOIC**, vytvorená pomocou aplikácie LOIC, v aplikácií je možné zvoliť protokol útoku, tu bol zvolený UDP, a veľkosť. Táto sada je ideálnou vzorkou pre prvotné experimentovanie, pretože napĺňa predpoklad, ktorý očakáva táto práca a to taký že v danej sade sú pakety útoku, ktoré majú sebe podobný payload.
- Ďalšie dve dátové sady sú: dátová sada **UDP** obsahujúca pakety, ktoré sú súčasťou DDoS útoku typu UDP flood a dátová sada **NTP**, ktorá obsahuje NTP amplifikačný DDoS útok. Zdroj týchto dátových sád je z datasetu DDoS Evaluation Dataset (CIC-DDoS2019)[1] od Canadian Institute for Cybersecurity.

Pre experimenty, ktoré budú spomenuté nižšie boli namiešavané sady určitých veľkostí. Pri tvorbe sád sa zakladalo na predpoklade, že pri DDoS útoku sú pakety patriace DDoS útoku zastúpené vo väčšine. Pre nižšie popísané experimenty bol zvolený pomer 30:70 (legitímna:nelegitímna). Každá sada sa teda skladala z 30% z paketov, ktoré boli náhodne vyberané zo sady LEGIT. Pakety útoku sa vyberali nasledovne:

- Malé sady z experimentu s každým pravidlom použitým samostatne, konkrétne sady s 30 a 130 paketmi, obsahujú pakety útoku zo sady LOIC. Sady s rovnakou veľkosťou z tohto experimentu, konkrétne s 5717 paketmi boli vytvorené, jedna z paketov zo sady LOIC a druhá pomocou paketov zo sady NTP.
- V ďalšom experimente bola použitá sada s 200 paketmi, v ktorej boli útočiace pakety náhodne vybrané z dvoch sád LOIC a UDP, pakety zo sady UDP boli ale zastúpené v menšom množstve a boli vyberané pakety, ktoré spĺňali predpoklad sebe podobnosti

[1] https://www.unb.ca/cic/datasets/ddos-2019.html

útočiacich paketov. Ďalšia použitá sada bola sada s 10000 paketmi, v ktorej sa nachádzali pakety NTP amplifikačného útoku. Taktiež sada s 50000 paketmi obsahovala pakety útoku zo sady NTP.

Pre experimenty boli sady každej danej veľkosti, spomínané vyššie, vytvorené dvakrát. Vždy s obsahom rôznych paketov ale toho istého typu a počtu. Sady sú trénovacia a testovacia. Na experiment je samozrejme treba zároveň aj sada s legitímnou prevádzkou. Takzvaná trénovacia sada sa používa na jednotlivé kroky metódy, popísané v časti 3.2, čiže vypočítanie štatistík a porovnávanie so štatistikou zo sady s legitímnou prevádzkou pomocou pravidiel. Výstupom tohto procesu sú n-gramy, vzory definujúce útok. Vyhodnotenie prebieha však na druhej sade, na takzvanej testovacej sade, v ktorej sú výsledné vzory následne hľadané v paketoch a označované za pozitívne sú pakety, ktoré sa skladajú aspoň z 20% z výsledných vzorov, ako bolo písané v časti 3.2. Následne sa porovnávajú označené pakety so skutočnosťou a vyhodnocujú sa výsledky.

## 5.2 Experimenty

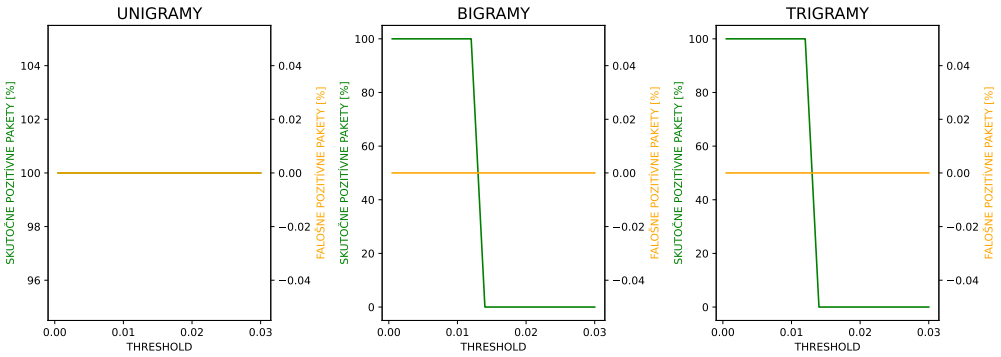
### Experiment s každým pravidlom použitým samostatne

Prvým vykonaným experimentom bol experiment, ktorý mal za cieľ overiť jednotlivé porovnávacie pravidlá samostatne. Pravidlá boli popísané v časti 3.2. Pri každom pravidle je skúmaný vplyv nastavovaného thresholdu pre dané pravidlo. Experiment má za cieľ ukázať ako sa mení kvalita odhaľovania vzorov na základe thresholdu a zároveň ukázať do akej miery jednotlivé pravidlá odhaľujú vzory ak by boli použité samostatne. Tento experiment bol vykonaný na viacerých odlišne veľkých dátových sadách. Z každej veľkosti boli vytvorené 3 sady. Sady boli porovnávané a vyhodnocované medzi sebou, vždy jedna sada obsahovala iba legitímnu prevádzku a druhá pomiešanú s útokom, takzvaná trénovacia sada. Pre evaluáciu výsledných vzorov bola použitá testovacia sada, ktorá bola namiešaná rovnako ako trénovacia ale obsahovala odlišné pakety. Pre tento experiment boli využité namiešané dátové sady, ktoré boli vytvorené kombináciou legitímnych paketov z dátovej sady LEGIT a z paketov ktoré patria útoku z dátových sád LOIC a prípadne NTP v pomere 30:70 (legitímne:nelegitímne). Dátové sady ktoré reprezentovali legitímnu prevádzku obsahovali rozdielne legitímne pakety ako boli použité v namiešanej dátovej sade s útokom. Prvé veľkosti dátových sád boli veľmi malé a slúžili na prvotné otestovanie metódy ako takej či funguje vôbec pre malú vzorku. Použitie pravidiel spoločne bude predmetom skúmania v ďalšom experimente.

Pre prvé pravidlo, ktoré porovnáva celkový počet výskytov konkrétneho n-gramu v jednotlivých štatistikách, boli skúmané výsledky na základe zmeny thresholdu, s ktorým sa porovnáva rozdiel počtu výskytov. Formálny zápis pravidla spolu s thresholdom bol znázornený v časti 3.2. Threshold bol zvolený a skúmaný v rozmedzí od 0 po 0.03, čo sa javilo ako vhodné rozmedzie na skúmanie po vypísaní a vyhodnotení výsledkov rozdielu dvoch pomerov v prvom pravidle. Rozmedzie pokrýva sledovanú zmenu priebehov skutočnej pozitivity a falošnej pozitivity. Po použití pravidla boli vzory, ktoré sú na výstupe metódy použité na evaluáciu a boli hľadané v paketoch. Pre rozhodovanie ktoré pakety označiť ako pozitívne bolo zisťované koľko percent daného paketu boli vzory. Pre účely všetkých experimentov bola hranica nastavená na 20%. Výsledok experimentu je znázornený grafmi. Grafy pre každú dátovú sadu sú tri, konkrétne pre každý typ skúmaných n-gramov, čiže unigramy, bigramy a trigramy. Na osi x každého grafu sú hodnoty zvolených thresholdov. Osi y sú dve, ľavá a pravá. Ľavá os je mierka pre zelenú krivku a znázorňuje koľko percent

paketov bolo na základe výsledných vzorov označených za pakety patriace útoku správne, takzvaný true positive rate. Pravá os je zasa mierka pre oranžovú krivku a znázorňuje koľko percent paketov bolo označených za útočiace nesprávne, takzvaný false positive rate.

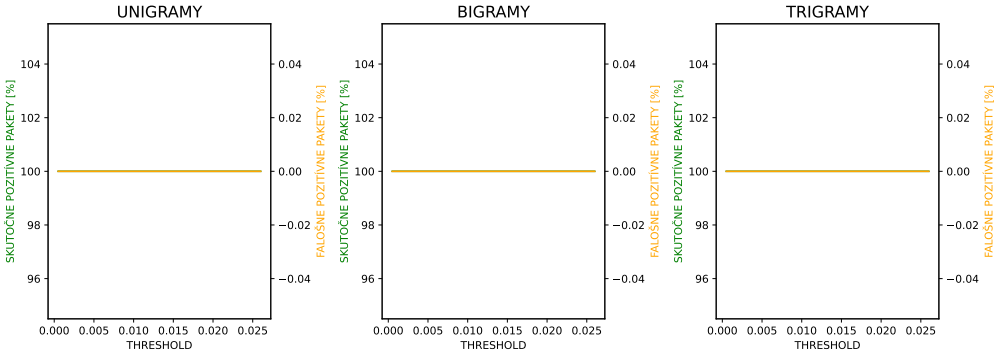
Ako prvé v tomto experimente bolo overenie funkčnosti navrhnutej metódy s použitím prvého pravidla na malých dátových sadách. Prvá namiešaná dátová sada obsahovala iba malú vzorku, spolu 30 paketov z toho 70% boli pakety útoku. Pakety útoku obsahovali rovnakú správu v payloade aby bola overená funkčnosť metódy na základe predpokladu sebe podobnosti payloadu paketov. Výsledok metódy na tejto dátovej sade je znázornený na grafoch na obrázku 5.1.




Obr. 5.1: Experiment s thresholdom v prvom pravidle na malej dátovej sade s 30 paketmi

Na grafoch 5.1 zelená krivka znázorňujúca správne označené pakety za útočiace, takzvané falošne pozitívne pakety, ukazuje že už pri zvolenom rozmedzí thresholdu je v prípade unigramov úspešnosť vysoká 100-percentná. Oranžová krivka, označujúca falošne pozitívne pakety, zároveň ukazuje že miera je nulová pre každý threshold. Výstup metódy s použitím prvého pravidla sa dá v tomto prípade považovať za úspešný. V prípade bigramov a trigramov je vidno, že miera skutočnej pozitivity pre thresholdy nižšie ako 0.012 je 100% a zároveň miera falošnej pozitivity je nulová, čo značí úspešné vybrané vzory na výstupe metódy. Na výstupe pre threshold 0.012 boli štyri bigramy a štyri trigramy. Ale pre thresholdy vyššie už úspešnosť miery skutočnej pozitivity paketov klesla na 0% a zároveň miera falošnej pozitivity zostala na 0%, z dôvodu, že na výstupe metódy neboli už žiadne vzory.

Ďalšou z malých dátových sád použitých na overenie funkčnosti metódy s použitím prvého pravidla bola namiešaná sada obsahujúca 130 paketov. Pakety útoku boli však odlišné ako v prvom prípade no stále sa zakladalo na predpoklade sebe podobnosti payloadu paketov. Výsledok metódy na tejto dátovej sade je znázornený na grafoch na obrázku 5.2.

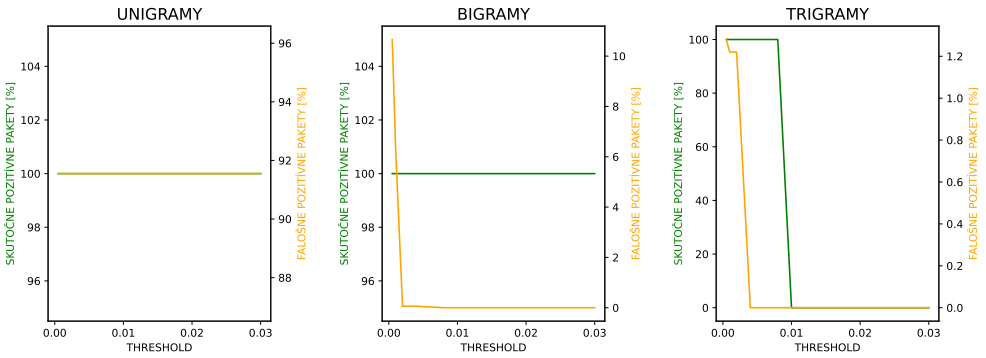



Obr. 5.2: Experiment s thresholdom v prvom pravidle na dátovej sade so 130 paketmi

Na grafoch 5.2 je vidno, že v namiešaných dátových sadách s paketmi boli týmto pravidlom nájdené vzory, ktoré úspešne definujú pakety útoku pre všetky thresholdy zo skúmaného rozmedzia. Výsledok sa teda dá považovať za úspešný. Toto platilo pre všetky typy, unigramy, bigramy a aj trigramy. Na vstupoch boli iba tie vzory ktoré definovali útok a nenachádzali sa v legitímnych paketoch, respektíve nenachádzali sa vo viac ako 20% žiadneho legitímneho paketu. Zelená krivka ukazuje 100% úspešnosť skutočne pozitívnych paketov a zároveň oranžová krivka znázorňuje nulovú mieru falošne označených paketov. Napríklad pre threshold 0.025 bolo na výstupe 16 unigramov, 4 bigramy a 4 trigramy definujúce útok.

Po overení funkčnosti tohto pravidla na malých dátových sadách sa funkčnosť overovala aj na sadách namiešaných z o niekoľko tisíc viac paketov. Zároveň sa overovala funkčnosť aj na sade namiešanej s útočiacimi paketmi zo sady LOIC a následne sa použila na namiešanie aj sada s útočiacimi paketmi zo sady NTP. Pakety z tejto sady obsahujú tiež rovnakú určitú časť payloadu podľa predpokladu, že si útočiace pakety sú sebe podobné. Výsledky sú znázornené na grafoch na obrázkoch 5.3 a 5.4.

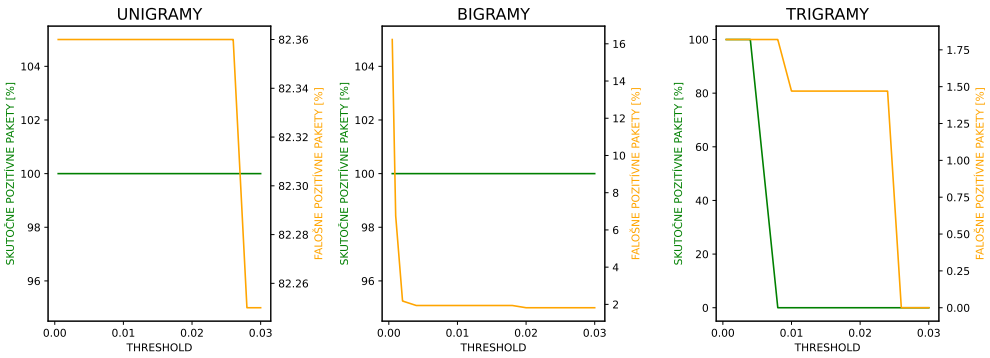
Namiešaná dátová sada z LOIC a legitímnej prevádzky obsahovala 5717 paketov z toho 70% boli pakety útoku. Pakety útoku obsahovali rovnakú správu v payloade aby bola overená funkčnosť metódy na základe predpokladu sebe podobnosti payloadu paketov. Výsledok metódy na tejto dátovej sade je znázornený na grafoch na obrázku 5.3.




Obr. 5.3: Experiment s thresholdom v prvom pravidle na dátovej sade s 5717 paketmi

Na grafoch na obrázku 5.3 je vidno, že metóda s prvým pravidlom pri väčších dátových sadách, označuje na výstupe aj viac vzorov. Medzi týmito vzormi sa však nachádza už aj

pomerne dosť vzorov, ktoré nedefinujú útočiacu prevádzku. To spôsobuje že stúpa miera falošne pozitívnych paketov, čo je znázornené oranžovou krivkou. Najviac sa prejavuje tento výsledok pri unigramoch, kde je síce miera skutočne pozitívnych paketov 100% pre všetky skúmané thresholdy, no zároveň je vysoká aj miera falošne pozitívnych paketov, ktorá je okolo 91,6%. Pri bigramoch je miera skutočne pozitívnych paketov 100% pre všetky skúmané thresholdy, a miera falošne pozitívnych paketov je už vyššia v porovnaní s menšími dátovými sadami, okolo 11%. To však platí iba v prípade malého thresholdu, menšieho ako 0.002. V prípade väčších thresholdov je táto miera zasa nulová. V prípade trigramov je miera skutočne pozitívnych paketov 100% pre thresholdy menšie ako 0.01. V tomto bode sa zelená krivka láme a pre vyššie thresholdy ukazuje mieru 0%. Toto bolo spôsobené tým že pre vyššie thresholdy neboli na výstupe označené žiadne vzory. Preto zároveň aj miera falošne pozitívnych paketov bola nulová. Tá sa pre thresholdy nižšie ako 0.004 držala okolo 1,2%, kedy bolo na výstupe 26 trigramov. Tento výsledok je veľmi dobrý, no dokonca pri zvýšení thresholdu až po 0.008, kedy boli na výstupe 4 trigramy definujúce útok, bola miera falošne pozitívnych paketov nulová.




Obr. 5.4: Experiment s thresholdom v prvom pravidle na dátovej sade s 5717 paketmi, obsahujúce pakety NTP amplifikačného útoku

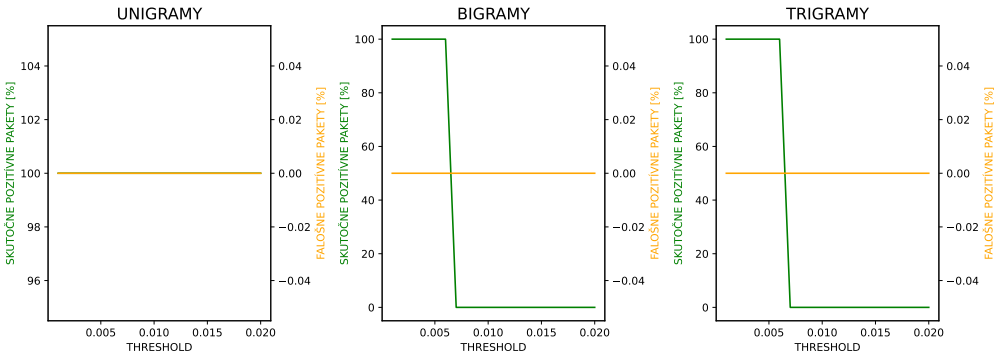
Na grafoch na obrázku 5.4 je zasa vidno, aké sú výsledky na rovnakej veľkosti dátovej sady ale s použitím paketov NTP amplifikačného útoku. V payloadoch týchto paketov sa nachádza určitá časť rovnaká pre všetky pakety, no ostatná väčšina payloadu je odlišná. Výsledky sú pomerne podobné ako pri vyššie spomínanej dátovej sade. Aj pri tejto metóde vidno vysokú mieru falošne pozitívnych paketov pri unigramoch. Pri bigramoch je miera skutočne pozitívnych paketov tiež 100% pre všetky skúmané thresholdy, a miera falošne pozitívnych paketov je tektiež vyššia v porovnaní s menšími dátovými sadami, okolo 16%. To však platí iba v prípade malého thresholdu, menšieho ako 0.002. V prípade väčších thresholdov je táto miera okolo 2%, na výstupe bolo 145 bigramov. V prípade trigramov je miera skutočne pozitívnych paketov 100% pre thresholdy menšie ako 0.005, kedy bolo na výstupe 112 trigramov. V tomto bode sa zelená krivka láme a pre vyššie thresholdy ukazuje mieru 0%. Toto bolo spôsobené tým že pre vyššie thresholdy bolo nájdených iba 10 vzorov, ktoré však nedefinovali útočiace pakety, ale zároveň sa nachádzali v niektorých legitímnych paketoch, čo spôsobilo mieru 1,5% falošnej pozitivity.

Pri výsledkoch skúmania prvého pravidla samostatne vidno, že metóda odhaľuje útočiacu prevádzku pomerne úspešne, a dá sa povedať že tento návrh je použiteľný v dátových sadách určitej veľkosti. No treba dodať fakt, ktorý experimenty ukázali, že s rastúcou veľkosťou dátovej sady začínajú byť kratšie n-gramy nevhodnejšie na použitie. V tomto prípade

to bolo vidno výrazne na unigramoch. Výsledok na základe miery skutočnej pozitivity a falošnej pozitivity ukazuje aj vhodné rozmedzie pre threshold prvého pravidla a to od 0.001 až po 0.015 prípadne po 0.02.

V nasledujúcej časti je popísaný experiment s druhým pravidlom samostatne, ktoré porovnáva počet paketov v ktorých sa konkrétny n-gram vyskytoval v jednotlivých štatistikách. Tak ako v experimente s prvým pravidlom samostatne, boli skúmané zmeny na základe thresholdu, s ktorým sa porovnáva rozdiel počtu výskytov v paketoch. Formálny zápis pravidla spolu s thresholdom bol znázornený v časti 3.2. Threshold bol zvolený a skúmaný v rozmedzí od 0.001 až po 0.02, s krokom 0.001, čo sa javilo ako vhodné rozmedzie na skúmanie po vypísaní a vyhodnotení výsledkov rozdielu dvoch pomerov v druhom pravidle. Rozmedzie pokrýva sledovanú zmenu priebehov skutočnej pozitivity a falošnej pozitivity. Po použití pravidla boli vzory, ktoré sú na výstupe metódy vyhodnocované rovnako ako v prípade prvého pravidla.

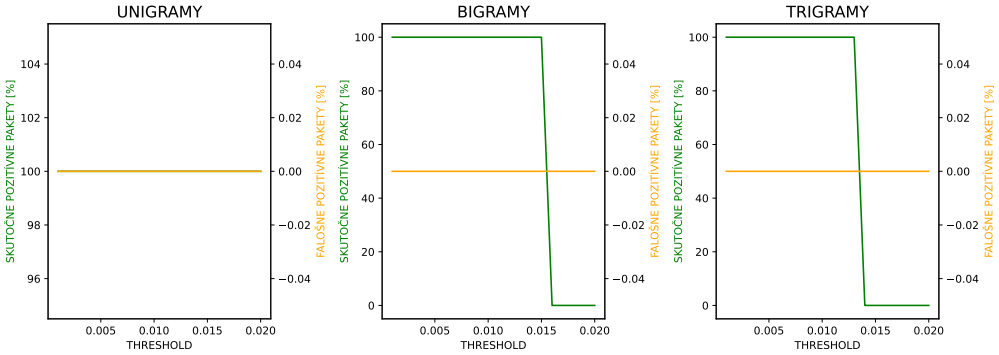
Ako prvé v tomto experimente bolo overenie funkčnosti navrhnutej metódy s použitím druhého pravidla na malých dátových sadách. Prvá namiešaná dátová sada obsahovala ako aj v prípade prvého pravidla, spolu 30 paketov z toho 70% boli pakety útoku. Namiešaná bola tiež rovnakým princípom, no z iných legitímnych paketov z dátovej sady LEGIT. Výsledok metódy na tejto namiešanej dátovej sade je znázornený na grafoch na obrázku 5.5.




Obr. 5.5: Experiment s thresholdom v druhom pravidle na dátovej sade s 30 paketmi

Na grafoch 5.5 zelená krivka, ukazuje že pri celom zvolenom rozmedzí thresholdu je v prípade unigramov úspešnosť vysoká 100-percentná. Oranžová krivka, zároveň ukazuje že miera je nulová pre každý threshold. Tieto hodnoty reflektujú to že na výstupe boli iba unigramy definujúce útok, konkrétne ich bolo šestnásť. V prípade bigramov, a identicky aj trigramov je zelená krivka na tom podobne iba do thresholdu 0.006, kedy sú na výstupe vzory, ktoré definujú iba pakety útoku, konkrétne 25 bigramov a 25 trigramov. Pre vyššie thresholdy už nie sú na výstupe žiadne vzory, preto obe krivky znázorňujú hodnoty 0.

Ďalšou z dátových sád použitých na overenie funkčnosti metódy s použitím druhého pravidla bola namiešaná sada obsahujúca 130 paketov. Pakety boli vybrané náhodne aby znázorňovali inú vzorku ako v prípade prvého pravidla, avšak pakety útoku rovnako obsahovali rovnakú správu v payloade aby bola overená funkčnosť metódy na základe predpokladu sebe podobnosti payloadu paketov. Výsledok metódy na tejto dátovej sade je znázornený na obrázku 5.6.

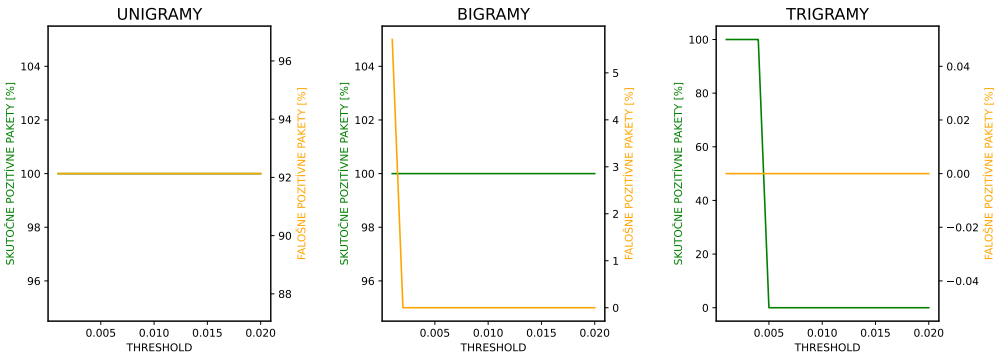



Obr. 5.6: Experiment s thresholdom v druhom pravidle na dátovej sade so 130 paketmi

Na grafoch 5.6 je vidno, že v namiešaných dátových sadách s paketmi boli týmto pravidlom nájdené vzory, ktoré definujú iba útočiace pakety, medzi nimi neboli žiadne zle vyhodnotené vzory, ktoré by falošne označili pakety a preto oranžová krivka zostáva vo všetkých troch typoch na 0% pre celé rozmedzie thresholdu. Miera skutočne pozitívnych paketov sa teda zároveň držala na hodnote 100%. No v prípade bigramov a trigramov postupne s navyšovaním thresholdu klesla na 0%, čo bolo spôsobené tým, že na výstupe už neboli žiadne vzory.

Po skúmaní funkčnosti tohto pravidla na malých dátových sadách sa funkčnosť overovala aj na sadách o niečo väčších. Zároveň sa overovala funkčnosť aj na sade namiešanej s útočiacimi paketmi zo sady LOIC a následne sa použila na namiešanie aj sada s útočiacimi paketmi zo sady NTP. Pakety z tejto sady obsahujú tiež rovnakú určitú časť payloadu podľa predpokladu, že si útočiace pakety sú sebe podobné. Výsledky sú znázornené na grafoch na obrázkoch 5.7 a 5.8.

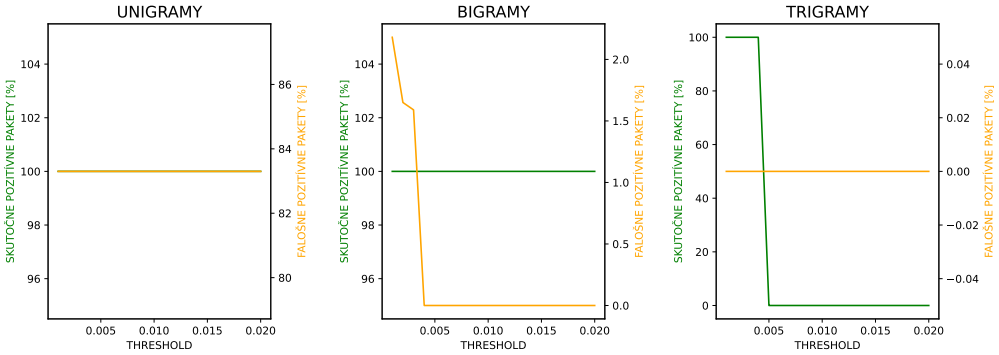
Namiešaná dátová sada z LOIC a legitímnej prevádzky obsahovala 5717 paketov z toho 70% boli pakety útoku. Pakety boli namiešané odlišné ako v prípade skúmania prvého pravidla. Pakety útoku obsahovali rovnakú správu v payloade aby bola overená funkčnosť metódy na základe predpokladu sebe podobnosti payloadu paketov. Výsledok metódy na tejto dátovej sade je znázornený na grafoch na obrázku 5.7.




Obr. 5.7: Experiment s thresholdom v druhom pravidle na dátovej sade s 5717 paketmi

Na grafoch na obrázku 5.7 je vidno, že aj pri metóde s využitím iba druhého pravidla nastáva pri väčších dátových sadách podobný problém ako v predošlom prípade keď bolo

skúmané využitie iba prvého pravidla. Tento problém je vidno na unigramoch, tie sú veľmi krátke n-gramy, po porovnávaní štatistík je tým pádom veľa z nich na výstupe označených za vzory. No okrem vzorov ktoré skutočne definujú útok je medzi nimi aj veľa takých, ktoré označujú pakety za útočiace falošne, to znamená vysokú mieru falošne pozitívnych paketov, v tomto prípade až 92%. V prípade bigramov je úspešnosť výstupu metódy podobná ako pri prvom pravidle, na výstupe sú pre thresholdy vyššie ako 0.002 vzory, konkrétne 25 bigramov, s úspešnosťou 100% a s nulovou falošnou pozitivitou. V prípade trigramov je miera skutočne pozitívnych paketov 100% pre thresholdy menšie ako 0.004. Na výstupe bolo 25 trigramov. Zároveň na výstupe neboli žiadne vzory, ktoré by zvyšovali falošnú pozitivitu ktorá teda zostala na 0%. V tomto bode sa zelená krivka láme a pre vyššie thresholdy ukazuje mieru 0%. Toto bolo spôsobené tým že pre vyššie thresholdy neboli na výstupe označené žiadne vzory. Preto zároveň aj miera falošne pozitívnych paketov bola nulová.




Obr. 5.8: Experiment s thresholdom v druhom pravidle na dátovej sade s 5717 paketmi, obsahujúce pakety NTP amplifikačného útoku

Na grafoch na obrázku 5.8 je zasa vidno, aké sú výsledky na rovnakej veľkosti dátovej sady ale s použitím paketov NTP amplifikačného útoku. V payloadoch týchto paketov sa nachádza určitá časť rovnaká pre všetky pakety, no ostatná väčšina payloadu je odlišná. Výsledky sú pomerne podobné ako pri vyššie spomínanej dátovej sade. Aj pri tejto metóde vidno vysokú mieru falošne pozitívnych paketov pri unigramoch. Pre bigramy je napríklad pri thresholde 0.004 miera falošnej pozitivity približne 1,5% a skutočnej pozitivity 100%, na výstupe bolo 87 bigramov. Pre trigramy je dokonca výsledok rovnaký ako v predošlej dátovej sade. Pre threshold bolo na výstupe 90 trigramov, definujúcich útok. Pre vyššie thresholdy na výstupe už neboli žiadne trigramy.

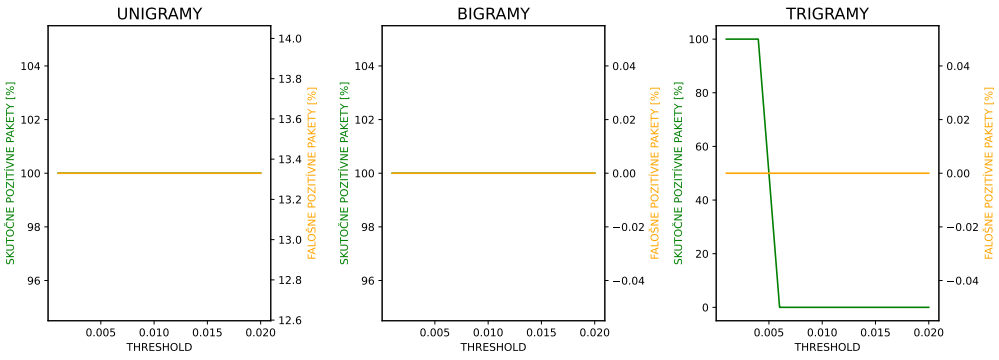
Pri výsledkoch skúmania druhého pravidla vidno, v podstate ten istý výsledok ako v prípade prvého pravidla. Metóda odhaľuje útočiacu prevádzku pomerne úspešne. No zároveň je potvrdené tiež, že s rastúcou veľkosťou sady sú výsledky kratších n-gramov horšie. Výrazne sa to ukázalo na unigramoch. Ako vhodné rozmedzie pre threshold sa javí nastavenie pre thresholdy do 0.01, na základe mier skutočnej a falošnej pozitivity.

Tento experiment mal za cieľ overiť použiteľnosť jednotlivých pravidiel metódy samostatne. V nasledujúcom experimente bude overená metóda na viacerých sadách s obidvoma pravidlami spoločne.

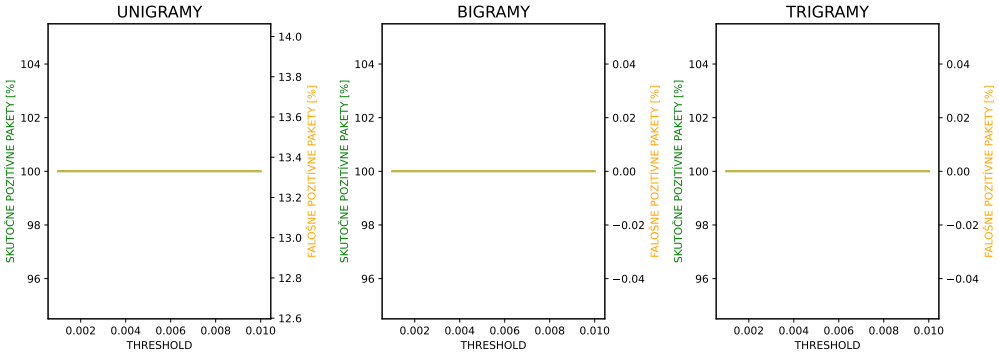
### Experiment s metódou s kombináciou pravidiel na dátových sadách

Tento experiment mal za cieľ overiť funkčnosť metódy na rôznych dátových sadách. V prvom experimente boli testované jednotlivé pravidlá použité v metóde samostatne. V tomto budú použité pravidlá už spoločne. Pravidlá boli popísané v časti 3.2. Po prvom experimente boli zvolené podľa výsledkov určité thresholdy. Experiment bol vykonaný tak, že najskôr bol zadaný threshold prvého pravidla na pevno a threshold druhého pravidla sa skúmal v určitom rozmedzí. Potom to prebiehalo naopak. Výber thresholdov sa zakladal na výsledkoch prvého experimentu na jednotlivých pravidlách, hladené bolo na vhodný threshold pre trigramy, aby nebol príliš vysoký a tým pádom by na výstupe metódy neboli žiadne trigramy. Zároveň aby bol tiež vhodný a nebol nízky pre bigramy a unigramy. Preto boli vybrané thresholdy 0.005 pre prvé pravidlo a 0.004 pre druhé. Tento experiment bol vykonaný na viacerých odlišne veľkých dátových sadách, porovnávaných pravidlami medzi sebou, vždy jedna sada obsahovala iba legitímnu prevádzku a druhá pomiešanú, takzvaná trénovacia sada. Následná evaluácia výsledných vzorov bola vykonaná na inej namiešanej, testovacej sade. Pre tento experiment boli využité namiešané dátové sady, ktoré boli vytvorené kombináciou legitímnych paketov z dátovej sady LEGIT a z paketov ktoré patria útoku z dátovej sady NTP a z dátových sád LOIC spolu s UDP. Pomer medzi paketmi bol 30:70 (legitímne:nelegitímne), ktorý sa zakladal na spomínanom predpoklade, že pakety patriace DDoS útoku sú zastúpené vo väčšine. Dátové sady ktoré reprezentovali legitímnu prevádzku obsahovali rozdielne legitímne pakety ako boli použité v namiešanej dátovej sade. Z použitých veľkostí na tento experiment, budú v tejto časti popísané tri. Menšia namiešaná sada obsahujúca približne 200 paketov, väčšia obsahujúca 10000 paketov a na záver bude overená použiteľnosť na dátovej sade s 50000 paketmi.

Ako prvé budú znázornené výsledky s dátovou sadou s 200 paketmi. Sada bola namiešaná z legitímnej sady a na útok boli použité pakety zo sady UDP a LOIC. Bola skúmaná funkčnosť metódy a výsledky sú znázornené na grafoch 5.9 a 5.10.



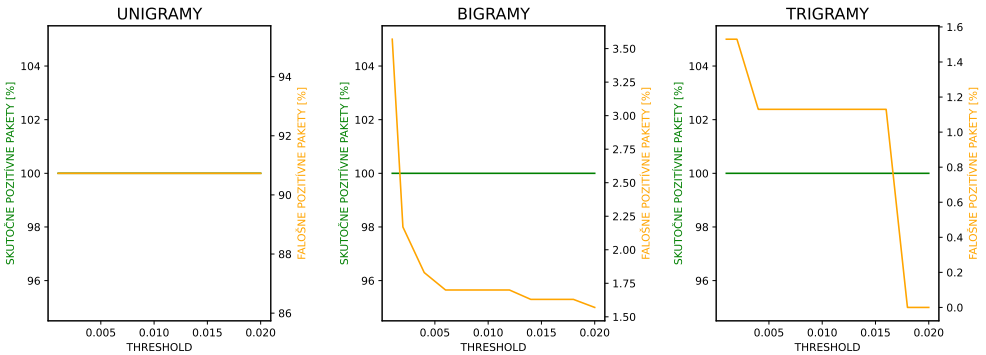

Obr. 5.9: Experiment so zmenou thresholdu prvého pravidla na dátovej sade s 200 paketmi



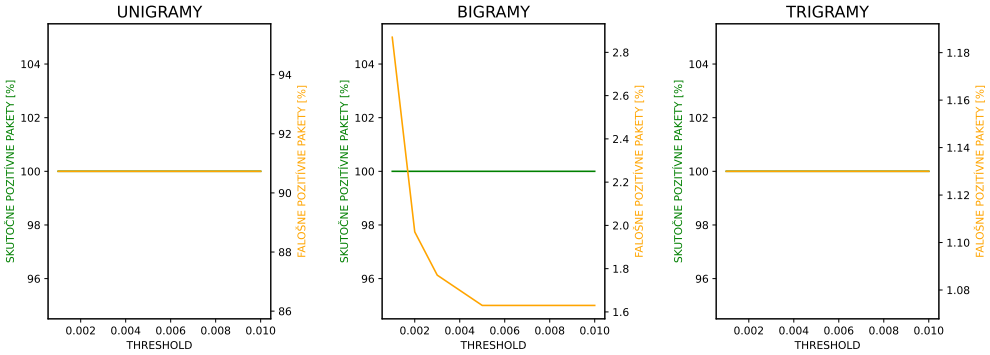

Obr. 5.10: Experiment so zmenou thresholdu druhého pravidla na dátovej sade s 200 paketmi

V prípade nastavenia thresholdu druhého pravidla napevno na hodnotu 0.004 je na grafoch na obrázku 5.9 vidno, že metóda má pre unigramy mieru skutočne pozitívnych paketov 100% no zároveň je miera falošnej pozitivity približne 13,4%. Túto mieru spôsobujú unigramy na výstupe, ktoré sa okrem paketov útoku nachádzali vo väčšej miere aj v 8 legitímnych paketoch. To isté sa stalo aj pri nastavení thresholdu prvého pravidla napevno, obrázok 5.10. Pre bigramy a trigramy je úspešnosť v dátovej sade tejto veľkosti dobrá, pakety ktoré mali byť označené ako útočiace takto označené aj boli. Zároveň miera falošne pozitívnych paketov je nulová. Avšak v prípade nastavenia thresholdu prvého pravidla napevno je vidno na zelenej krivke že klesla. Pri thresholde 0.004 boli ešte zvolené 4 trigramy, ktoré definovali pakety útoku no pri thresholde 0.006 bol na výstupe už iba jeden trigram a ten sa nenachádzal v útočiacich paketoch v takej miere aby boli označené za útočiace. Preto obe krivky v tomto prípade mali hodnotu 0%.

Ďalšou skúmanou sadou bola napríklad sada s veľkosťou 10000 paketov. Táto sada bola namiešaná z legitímnych paketov a paketov s NTP útokom. Na tejto dátovej sade sa overovala funkčnosť metódy a výsledky sú znázornené na grafoch na obrázku 5.11 a 5.12.



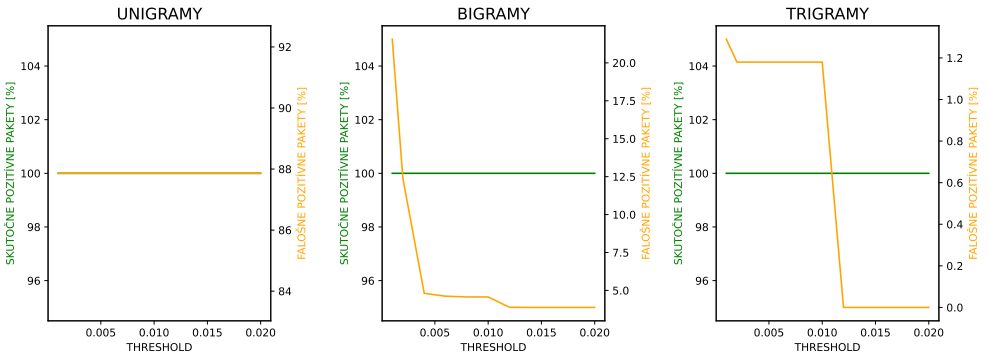

Obr. 5.11: Experiment so zmenou thresholdu prvého pravidla na dátovej sade s 10000 paketmi, obsahujúce pakety NTP amplifikačného útoku



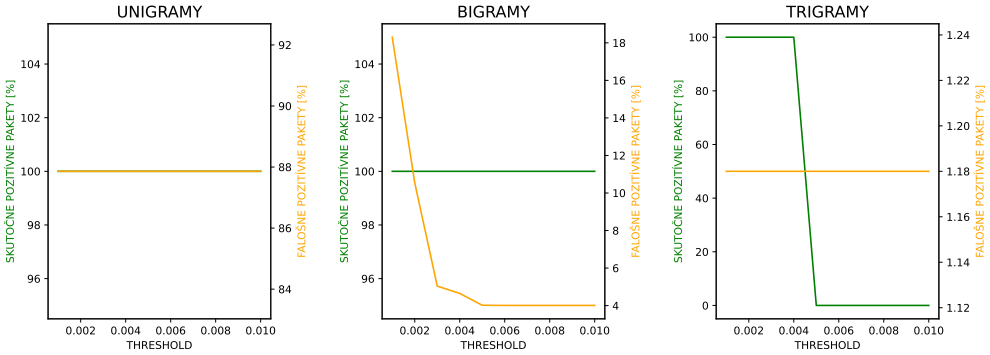

Obr. 5.12: Experiment so zmenou thresholdu druhého pravidla na dátovej sade s 10000 paketmi, obsahujúce pakety NTP amplifikačného útoku

V prípade nastavenia thresholdu druhého pravidla napevno na hodnotu 0.004 je na grafoch na obrázku 5.11 vidno, že metóda nieje efektívna pri použití unigramov. Pri nich je miera falošnej pozitivity vysoká, pretože medzi výslednými vzormi bolo veľa zle vyhodnotených. Pre bigramy a trigramy je úspešnosť v dátovej sade tejto veľkosti pomerne dobrá, pakety ktoré mali byť označené ako útočiace takto označené aj boli. Zároveň je nízka miera falošne pozitívnych paketov. Napríklad pri nastavení thresholdu na 0.015 je na výstupe 108 trigramov, no pri nastavení thresholdu na 0.018 je na výstupe 107 trigramov. Čo ukazuje že mieru falošnej pozitivity približne 1,2% v tomto prípade spôsoboval iba jeden trigram, konkrétne trigram `ffffff` v hexadecimálnej forme. Výsledky sú podobné aj pri nastavení hodnoty thresholdu prvého pravidla napevno a skúmaní thresholdu druhého pravidla v určitom rozsahu, znázornené na obrázku 5.12. Miera falošne pozitívnych paketov pri trigramoch je konštantná, no približne rovnaká ako v prvom prípade.

Metóda bola taktiež skúmaná na dátovej sade s 50000 paketmi. Táto sada bola namiešaná z legitímnych paketov a paketov s NTP útokom. Výsledky skúmania so zmenou thresholdu sú znázornené na grafoch na obrázku 5.13 a 5.14.




Obr. 5.13: Experiment so zmenou thresholdu prvého pravidla na dátovej sade s 50000 paketmi




Obr. 5.14: Experiment so zmenou thresholdu druhého pravidla na dátovej sade s 50000 paketmi

V prípade nastavenia thresholdu druhého pravidla napevno na hodnotu 0.004 je na grafoch na obrázku 5.13 vidno, že metóda na takto veľké dátové sady nieje efektívna pri použití unigramov, tie sú moc krátke keďže sa jedná o jednobajtové sekvencie. Pri nich je miera falošnej pozitivity vysoká, pretože medzi výslednými vzormi bolo veľa zle vyhodnotených. Úspešnosť bigramov tiež klesá. Miera falošne pozitívnych paketov je pre menšie thresholdy z vybraného rozsahu, menšie ako 0.004 vysoká a dosahuje 12 až 22%. Vyššie skúmané thresholdy už obmedzujú mieru falošnej pozitivity na hodnotu okolo 4%. Bigramy sa tiež javia pri takýchto veľkostiach sady ako krátke na použitie. S navýšením sady by začala miera falošnej pozitivity tiež stúpať ako pri unigramoch. Výsledky pre unigramy a bigramy sú podobné aj pri nastavení thresholdu prvého pravidla napevno na hodnotu 0.005. Pre trigramy bola v prvom prípade pre thresholdy vyššie ako 0.01 úspešnosť 100%. V druhom prípade zasa vidno že threshold vyšší ako 0.004 už obmedzil výstupné vzory metódy tak, že na výstupe boli iba vzory, konkrétne ich bolo 22, ktoré označovali falošné pakety a miera skutočne pozitívnych paketov bola nulová.

Výsledok skúmania metódy v tomto experimente ukázal zmenu mier falošne a skutočne pozitívnych paketov na základe zmeny thresholdov pre jednotlivé pravidlá a naznačil teda vhodné nastavenie thresholdov jednotlivých pravidiel. Cieľom dobrého nastavenia thresholdov je aby neobmedzili výstupné n-gramy, ktoré sú skutočne vzormi, no súčasne aby sa nezdvihla miera falošnej pozitivity. Threshold pre druhé pravidlo by mal byť nižší ako pre prvé. Zároveň je treba nastaviť thresholdy tak aby neboli príliš vysoké pre vyhodnotenie trigramov. Ďalším výsledkom tohto experimentu na viacerých dátových sadách je, že princíp metódy je použiteľný no s rastúcou veľkosťou dátovej sady úspešnosť krátkych n-gramov klesá. Unigramy, keďže sú to iba jednotlivé bajty sa pri väčšom počte paketov nedajú použiť, pretože majú vysokú mieru falošnej pozitivity. Táto miera stúpa aj pri bigramoch, ak sa navyšuje počet paketov. Pre lepšie výsledky metódy je teda vhodné použiť dlhšie n-gramy, kvôli priestorovej náročnosti a ako prvotný prístup boli však v tejto práci skúmané iba maximálne trigramy. Ako bolo spomínané v pokračovaní a nadviazaní na túto prácu bude hľadaný spôsob akým by sa dali efektívne využiť dlhšie n-gramy. Zároveň problémom ktorý môže metóda mať sú legitímne pakety, ktoré obsahujú payload obsahujúci iba jeden znak. Následne vysoký výskyt tohto znaku spôsobí označenie tohto vzoru a zároveň zvýšenie miery falošnej pozitivity.

# Kapitola 6

# Záver

Cieľom tejto práce bolo navrhnúť metódu na odvodzovanie vzoru pre mitigáciu DDoS útoku. Metóda sa zameriava na odvodzovanie vzoru z payloadu paketov, ktoré sú majoritne zastúpené v DDoS útokoch. Odvodený vzor by mal slúžiť a pomôcť pri následnej mitigácií zmierniť prebiehajúci útok. Cieľom tejto práce zároveň ale nebolo navrhnúť metódu, ktorá by prispela k blokovaniu všetkých typov DDoS útokov, ale k blokovaniu takých typov DDoS útokov, ktoré nejdú blokovať inak ako na základe odvodeného vzoru z payloadu.

Na účely navrhnutia metódy bola ako prvá naštudovaná relevantná literatúra, s ktorou problematika práce súvisí, popísaná v kapitole 2. Potrebné bolo naštudovať literatúru ktorá popisuje DDoS útoky a vlastnosti týchto útokov, ich typy, priebeh a využitie botnetu v ich prospech. Taktiež boli naštudované a popísané techniky mitigácie objemových DDoS útokov, ktoré boli relevantné pre cieľ tejto práce.

V kapitole 3 bolo navrhnutých a popísaných niekoľko rôznych prístupov pre hľadanú metódu. Nakoniec sa ukázalo, že každý z navrhovaných prístupov má svoje nevýhody. V tejto kapitole bol zvolený a popísaný spolu s dôvodmi zvolenia a jednotlivými krokmi, jeden z prístupov, ktorý bol použitý na implementáciu metódy. Konkrétne prístup využívajúci n-gramy. K n-gramom bolo potrebné pre následnú implementáciu naštudovanie nevyhnutnej literatúry, popísanej v kapitole 2.

V kapitole 4 bol následne popísaný spôsob implementácie jednotlivých krokov pracovného postupu navrhnutej metódy, znázornených v kapitole 3 navrhnutej metódy. Popísané boli použité nástroje. Výsledkom je program v podobe skriptu naprogramovaný v jazyku Python. Výstupom programu za predpokladu správneho zvolenia nastavení sú vzory nájdené v payloadoch paketov. Správne nastavenia boli predmetom skúmania metódy, popísané boli v kapitole 5. V tejto kapitole boli popísané vykonané experimenty a znázornené aj príslušnými grafmi. Experimenty boli vykonané, takzvane offline na dátových sadách simulujúcich DDoS útok. Na experimentoch bola skúmaná úspešnosť a použiteľnosť navrhnutej metódy.

V práci bolo zvažovaných viacero možných metód riešenia problematiky a zároveň bola implementovaná jedna. Čiastočné zhodnotenie ukazujú jednotlivé experimenty a ich výsledky. Celkový výsledok práce ukázal že navrhnutá metóda je použiteľná na problematiku tejto práce. Keďže sa však jedná o prvú fázu experimentov tak pokračovanie práce bude smerovať k rozširovaniu metódy o použitie dlhších n-gramov, vyriešenie priestorovej náročnosti ukladania týchto n-gramov a experimenty budú rozšírené o ďalšie dátové sady.

# Literatúra


[1] *Arbor Threat Mitigation System (TMS)* [online]. NETSCOUT [cit. 2022-05-01]. Data Sheet. Dostupné z: https://www.netscout.com/sites/default/files/2019-09/SECPDS_004_EN-1901%20-%20Arbor%20Threat%20Mitigation%20System%20%28TMS%29.pdf.

[2] Alzahrani, S. a Hong, L. Generation of DDoS Attack Dataset for Effective IDS Development and Evaluation. *Journal of Information Security*. 2018, zv. 9, č. 4, [cit. 2022-03-11]. Dostupné z: https://www.scirp.org/journal/paperinformation.aspx?paperid=86682.

[3] Amir, A., Charalampopoulos, P., Pissis, S. P. a Radoszewski, J. Dynamic and Internal Longest Common Substring. 2020, zv. 82, s. 3707–3743, [cit. 2022-04-24]. DOI: https://doi.org/10.1007/s00453-020-00744-0.

[4] Banerjee, S. a Pedersen, T. The Design, Implementation, and Use of the Ngram Statistics Package. In:. Február 2003, sv. 2000, s. 370–381 [cit. 2022-04-03]. ISBN 978-3-540-00532-2. Dostupné z: https://www.researchgate.net/publication/221629332_The_Design_Implementation_and_Use_of_the_Ngram_Statistics_Package.

[5] Bieganski, Riedl, Cartis a Retzel. Generalized suffix trees for biological sequence data: applications and implementation. In: *1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*. 1994, sv. 5, s. 35–44 [cit. 2022-04-24]. DOI: 10.1109/HICSS.1994.323593.

[6] Braden, R. T. *Requirements for Internet Hosts - Communication Layers* [RFC 1122]. RFC Editor, október 1989 [cit. 2022-04-21]. DOI: 10.17487/RFC1122. Dostupné z: https://www.rfc-editor.org/info/rfc1122.

[7] Cao, X., Li, S. a Tung, A. Indexing DNA Sequences Using q-Grams. In:. Apríl 2005, sv. 3453 [cit. 2022-04-03]. ISBN 978-3-540-25334-1. Dostupné z: https://www.researchgate.net/publication/220787574_Indexing_DNA_Sequences_Using_q-Grams.

[8] Charalampopoulos, P., Kociumaka, T., Pissis, S. P. a Radoszewski, J. Faster Algorithms for Longest Common Substring. In: *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, sv. 204 [cit. 2022-04-24]. Leibniz International Proceedings in Informatics (LIPIcs). DOI: 10.4230/LIPIcs.ESA.2021.30. ISBN 978-3-95977-204-4. Dostupné z: https://drops.dagstuhl.de/opus/volltexte/2021/14611.

[9] DNS Amplification Attacks. [online]. CISA. Marec 2013, [cit. 2022-03-19]. Dostupné z: https://www.cisa.gov/uscert/ncas/alerts/TA13-088A.

[10] Cooper, S. What is a botnet and how to avoid being part of one. [online]. Comparitech. Február 2018, [cit. 2022-03-09]. Dostupné z: https://www.comparitech.com/blog/information-security/what-is-a-botnet/.

[11] Guo, F., Chen, J. a Chiueh, T. Spoof Detection for Preventing DoS Attacks against DNS Servers. In: *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. 2006 [cit. 2022-03-30]. Dostupné z: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1648824.

[12] IP Spoofing. [online]. Imperva. [cit. 2022-03-29]. Dostupné z: https://www.imperva.com/learn/ddos/ip-spoofing/.

[13] Jacko, D. *Odvozování pravidel pro mitigaci DDoS útoků*. Brno, CZ, 2021. [cit. 2022-04-16]. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.fit.vut.cz/study/thesis/23920/.

[14] Keary, T. Dos vs DDoS Attacks: The Differences and How To Prevent Them. [online]. Comparitech. Január 2022, [cit. 2022-03-06]. Dostupné z: https://www.comparitech.com/net-admin/dos-vs-ddos-attacks-differences-prevention/.

[15] Kumar, S. Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet. In: *Second International Conference on Internet Monitoring and Protection (ICIMP 2007)*. 2007 [cit. 2022-03-30].

[16] Litwin, W., Mokadem, R., Rigaux, P. a Schwarz, T. Pattern Matching Using n-gram Sampling Of Cumulative Algebraic Signatures : Preliminary Results. *Lecture Notes in Computer Science - LNCS*. Január 2006, [cit. 2022-04-03]. Dostupné z: https://www.researchgate.net/publication/247122032_Pattern_Matching_Using_n-gram_Sampling_Of_Cumulative_Algebraic_Signatures_Preliminary_Results.

[17] Mahjabin, T., Xiao, Y., Sun, G. a Jiang, W. *A survey of distributed denial-of-service attack, prevention, and mitigation techniques* [online]. 2017 [cit. 2022-04-10]. Dostupné z: https://journals.sagepub.com/doi/pdf/10.1177/1550147717741463.

[18] Matoušek, P. *Síťové služby a jejich architektura*. Publishing house of Brno University of Technology VUTIUM, 2014 [cit. 2022-04-21]. 396 s. ISBN 978-80-214-3766-1. Dostupné z: https://www.fit.vut.cz/research/publication/10567.

[19] Muhammad Aminu, L., Shaikh, R. a Hassan, R. An Anomaly Mitigation Framework for IoT Using Fog Computing. *Electronics*. September 2020, zv. 9, s. 1565, [cit. 2022-03-08]. Dostupné z: https://www.researchgate.net/publication/350334702_A_DDoS_Attack_Mitigation_Framework_for_IoT_Networks_using_Fog_Computing.

[20] Patrikakis, C., Masikos, M. a Zouraraki, O. Distributed Denial of Service Attacks. *The Internet Protocol Journal* [online]. 2004, zv. 7, č. 4, [cit. 2022-03-11]. Dostupné z: https://web.archive.org/web/20190826143507/https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-30/dos-attacks.html.

[21] Ravoof, S. What is IP spoofing? [online]. Kinsta. Január 2022, [cit. 2022-03-30]. Dostupné z: https://kinsta.com/blog/ip-spoofing/.

[22] Singh, K. J. a De, T. DDOS Attack Detection and Mitigation Technique Based on Http Count and Verification Using CAPTCHA. In: *2015 International Conference on Computational Intelligence and Networks*. 2015 [cit. 2022-04-11]. Dostupné z: https://ieeexplore.ieee.org/document/7053830.

[23] How Firewalls Mitigate Attacks. In:. UKEssays, November 2018 [cit. 2022-04-11]. Dostupné z: https://www.ukessays.com/essays/computer-science/how-firewalls-mitigate-attacks-computer-science-essay.php?vref=1.

[24] Walkowski, D. What Is a Distributed Denial-of-Service Attack? [online]. Jún 2019, [cit. 2022-03-09]. Dostupné z: https://www.f5.com/labs/articles/education/what-is-a-distributed-denial-of-service-attack-.

[25] Walkowski, D. What Is a DNS Amplification Attack? [online]. Jún 2019, [cit. 2022-03-19]. Dostupné z: https://www.f5.com/labs/articles/education/what-is-a-dns-amplification-attack-.

[26] NTP amplification DDoS attack. [online]. Cloudflare. [cit. 2022-03-20]. Dostupné z: https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/.

[27] What is a DDoS attack? [online]. Cloudflare. [cit. 2022-03-14]. Dostupné z: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/.

[28] What is a WAF? | Web Application Firewall explained. [online]. Cloudflare. [cit. 2022-04-12]. Dostupné z: https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/.