VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

MITIGACE DOS ÚTOKŮ S VYUŽITÍM NEURONOVÝCH SÍTÍ

MITIGATION OF DOS ATTACKS USING NEURAL NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **TOMÁŠ ODEHNAL**
AUTHOR

VEDOUCÍ PRÁCE **JAN KUČERA, Ing.**
SUPERVISOR

BRNO 2018

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Odehnal Tomáš**
Program: Informační technologie
Název: **Mitigace DoS útoků s využitím neuronových sítí**
Mitigation of DoS Attacks Using Neural Networks
Kategorie: Počítačové sítě

Zadání:

1. Seznamte se s problematikou útoků typu odepření služby a zařízením vyvíjeným v rámci CESNET pro ochranu před těmito útoky.
2. Implementujte vybranou metodu pro potlačení DoS útoku zneužívajícího protokol TCP jako rozšiřující modul tohoto zařízení.
3. Nastudujte základy teorie konvolučních a rekurentních neuronových sítí a možnosti jejich uplatnění pro mitigaci DoS útoků.
4. Na základě dostupné literatury vyberte nebo navrhněte alternativní heuristickou metodu mitigace DoS využívající neuronovou síť.
5. Zvolenou metodu s využitím vhodných nástrojů implementujte a proveďte experimenty nad dostupnou datovou sadou.
6. Porovnejte dosažené výsledky a diskutujte možnosti dalšího pokračování práce.

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 http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Kučera Jan, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 26. října 2018

Abstrakt

Bakalářská práce se zabývá návrhem a implementací dvou přístupů pro ochranu před SYN Flood útoky, které patří do DoS útoků. Denial of Service útoky jsou v dnešní době velice rozšířené a jejich provedení není příliš náročné. Přitom mohou způsobit velké finanční škody ať už poskytovatelům připojení, nebo provozovatelům služeb. Cílem práce je zjistit, jestli konvenční algoritmický přístup a heuristický přístup s využitím neuronových sítí jsou schopny mitigace SYN Flood útoků. Podle návrhů obou přístupů byly provedeny jejich implementace. Obě byly následně otestovány.

Abstract

This bachelor's thesis deals with design and implementation of two approaches as protection against SYN Flood attacks, which are part of DoS attacks. Nowadays Denial of Service attack are very widespread and their execution are quite simple. While they can cause big financial damage to internet or service providers. The purpose of this work is to determine that conventional algorithmic approach and heuristic approach using neural network are capable of SYN Flood attacks mitigation. Implementation of both approaches were done by their design. Then both implementations were tested.

Klíčová slova

SYN Flood útok, odepření služby, Ochrana před útoky, ACK Spoofing, neuronové sítě, hluboké učení, DoS mitigace útoků

Keywords

SYN Flood attack, denial of service, protection against attacks, ACK Spoofing, neural network, deep learning, DoS attacks mitigation

Citace

ODEHNAL, Tomáš. *Mitigace DoS útoků s využitím neuronových sítí.* Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Jan Kučera, Ing.

Mitigace DoS útoků s využitím neuronových sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Kučery. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Odehnal
13. května 2019

Poděkování

Rád bych poděkoval vedoucímu Ing. Janu Kučerovi za odborné vedení, trpělivost, poskytnutí množství cenných rad, připomínek a návrhů při tvorbě této bakalářské práce. Dále bych chtěl poděkovat rodině a kamarádům za jejich podporu během bakalářského studia.

Obsah

Kapitola 1

Úvod

V posledních letech došlo k velkému technickému pokroku a informační technologie se staly naší každodenní součástí. Běžně nosíme u sebe chytré telefony, používáme své počítače k nejrůznějším aktivitám. K rozvoji došlo i v oblasti počítačových sítích, jejichž součástí je i velice využívaný Internet. Ten nám umožňuje používání služeb jako jsou např. posílání e-mailových zpráv, sdílení fotografií, prohlížení webových stránek nebo internetové bankovnictví. Avšak s rozšířením počítačových sítí došlo i k rozvoji kybernetických útoků, které mají za cíl např. odcizení osobních a soukromých dat, přihlašovací údaje k internetovému bankovnictví nebo odepření určité služby. Tyto útoky patří k nejčastějším kybernetickým útokům v počítačových sítích.

Útok typu odepření služby se označuje jako DoS (Denial of Service). Existuje i jejich distribuovaná varianta (DDoS). Jejich úlohou je zejména odepření přístupu legitimním uživatelům k určitým službám (např. webové stránky). Tato práce je zaměřena právě na obranu před těmito útoky. Především se zaměřuje na útoky typu SYN Flood. Tento útok zneužívá protokolu TCP, který je součástí transportní vrstvy modelu referenčního modelu ISO/OSI. V práci budou požity dva přístupy pro mitigaci útoku. První je konvenční algoritmický přístup, který stejně jako SYN Flood útok využívá chování protokolu TCP a označuje se jako ACK Spoofing (Acknowledge Spoofing). Druhý přístup je mitigace pomocí neuronových sítí, které se v dnešní době v hojném počtu využívají v různých oborech. Využití je např. v lékařství, ve finanční sféře , rozpoznávání textů, převod mluvené řeči do písemné podoby. Cílem práce je snaha zjistit, jestli oba přístupy lze použít právě pro mitigaci útoků. Zejména pak u neuronové sítě je potřeba zjistit, jestli nejen správně klasifikuje útok, ale i legitimní provoz.

V kapitole 2. je čtenáři poskytnut nutný teoretický základ, který je potřebný k pochopení dalších částí práce. Je zde popsán princip počítačových sítí, především protokol TCP. Dále jsou zde popsány nejznámější útoky, které tento protokol zneužívají, především SYN Flood, a obrana proti nim. Následně je zde popsáno, co jsou neuronové sítě, jakým způsobem fungují a jsou zde uvedeny jejich nejznámější typy.

V 3. kapitole je vysvětlen návrh konvenčního algoritmického přístupu i model neuronové sítě. Také jsou zde uvedeny požadavky na formu vstupních dat do neuronové sítě.

V kapitole 4. jsou popsány implementační detaily. Je zde jak prototypová, tak i modulová implementace ACK Spoofingu. Následně jsem popsal implementace programu na extrakci a úpravu dat pro neuronovou síť. A jako poslední je zde implementace samotné neuronové sítě.

V kapitole 5. jsou uvedeny dosažené výsledky. Je zde popsán způsob testování konvenčního algoritmického přístupu. Dále se zde nachází výsledky experimentů nad implementovanou neuronovou sítí.

Poslední 6. kapitola shrnuje celkové dosažené výsledky práce a obsahuje možnosti na její další možné budoucí směřování.

Kapitola 2

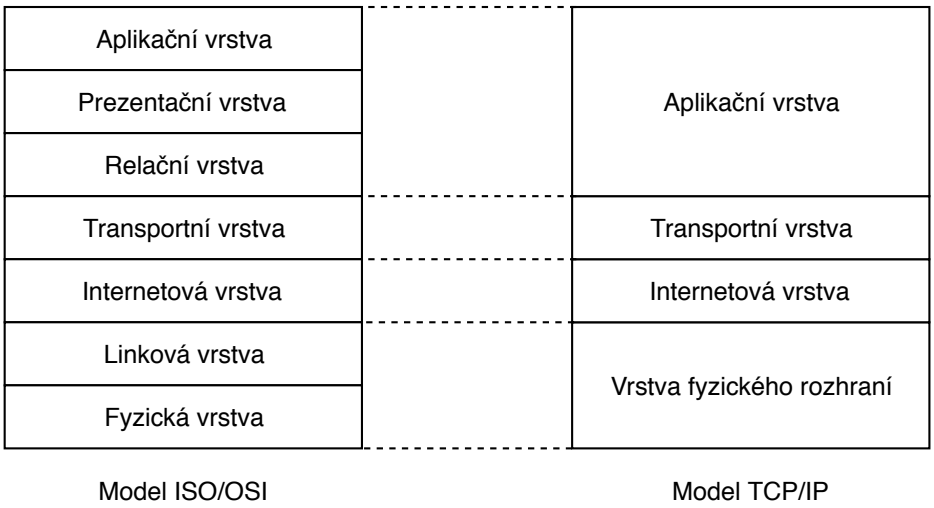
Teoretický rozbor

Následující kapitola obsahuje shrnutí teoretických informací potřebných pro řešení této práce.

2.1 Principy počítačových sítí

Pokud není uvedeno jinak, následující kapitola čerpá z [11].

Činnost a komunikace síťových služeb jsou dány architekturou počítačové sítě. V současné době se v oblasti Internetu používají dva modely architektury – model ISO/OSI, který slouží jako referenční model a model TCP/IP založený na protokolech TCP (Transmission Control Protocol) a IP (Internet Protocol). Referenční model OSI tvoří sedm vrstev, které definují služby příslušné vrstvy a odpovídající protokoly.



Obrázek 2.1: Porovnání referenčního modelu ISO/OSI a TCP/IP.

Implementace architektury TCP/IP je rozdělena do tří částí, jak lze vidět na obrázku 2.1. Nejnižší část je vrstva fyzického rozhraní, která popisuje standardy pro fyzické medium a elektrické signály. Tato vrstva také zahrnuje funkce pro přístup k fyzickému mediu (síťové rozhraní počítače). Na fyzické vrstvě se adresuje pomocí tzv. fyzické adresy, které jedno-

značně identifikuje síťové rozhraní v lokální síti. V případě technologie Ethernet se jedná o 48-bitovou adresu. Tato vrstva má také za úkol zapouzdření IP datagramů do rámců.

Nad vrstvou fyzického rozhraní se nachází internetová vrstva. Slouží k vytváření datagramů, adresování počítačů a směrování na místo určení. K identifikaci počítače slouží IP adresa. Tato adresa je ve verzi 4 protokolu IP 32-bitová a ve verzi 6 je 128-bitová. IP adresa je jedinečná v celém Internetu, oproti tomu fyzická adresa je jedinečná pouze v lokální síti. Vytváří tzv. logické spojení mezi počítači.

Další vrstvou je transportní vrstva, která slouží k identifikaci aplikací (služeb) na zařízeních (počítačích). Zajišťuje přenášení dat z aplikace na zdrojovém počítači do aplikace na cílovém počítači. Tedy protokoly transportní vrstvy zajišťují logické spojení mezi aplikačními procesy běžícími na (různých) počítačích. Data z aplikační vrstvy zapouzdřuje do paketů. Pro jednoznačnou identifikaci služeb se používá 16-bitové číslo portu. Základní protokoly transportní vrstvy jsou protokoly TCP a UDP. V této práci se budu věnovat pouze protokolu TCP, který je popsán dále v této kapitole.

Poslední vrstvou je aplikační vrstva, která je tvořena procesy a aplikacemi, které komunikují po síti. Zajišťuje zpracování dat na nejvyšší úrovni včetně reprezentace dat i kódování. Adresování na aplikační vrstvě záleží na konkrétní aplikaci (službě) a tedy i konkrétním aplikačním protokolu. Například u elektronické pošty se používá adresa ve formátu `user@host` (např. login123@seznam.cz).

Protokol UDP (User Datagram Protocol) je jednodušší než protokol TCP. Spojení protokolu UDP je jednodušší. UDP nepotřebuje ustanovit spojení, dále je komunikace mezi stanicemi bezstavová. Také oproti protokolu TCP nezajišťuje spolehlivý přenos.

Protokol TCP (Transmission Control Protocol) zajišťuje spolehlivý přenos dat z jednoho koncového zařízení na druhé koncové zařízení.

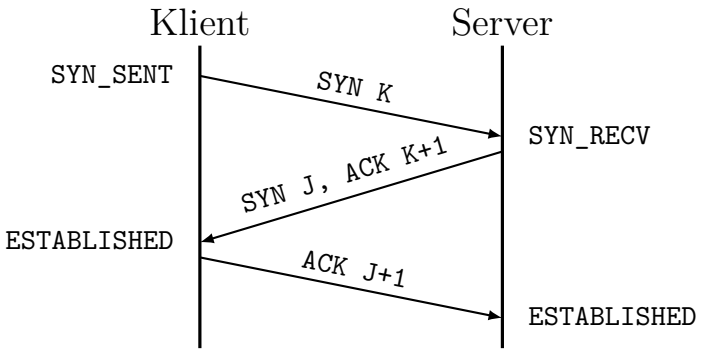
TCP vytváří spojení pomocí mechanismu třífázového spojení, angl. three-way handshake, který lze vidět na obrázku 2.2. Název je odvozen od způsobu komunikace (navazování spojení) mezi klientem a serverem, kteří si mezi sebou pošlou tři pakety:

- Klient pošle serveru TCP synchronizační paket s příznakem SYN (tzv. SYN paket).
- Server SYN paket příjme a odešle zpět klientovi TCP schvalovací paket s příznaky SYN a ACK (tzv. SYN-ACK paket).
- Klient příjme SYN-ACK paket a pošle zpět serveru TCP paket s příznakem ACK (tzv. ACK paket).

V obrázku je využita i architektura klient-server, což je standardní schéma komunikace mezi dvěma procesy. Procesy mohou běžet na stejném počítači.

2.2 Útoky typu odepření služby

S rozšířením internetu se během posledních několik let rozšířily i útoky. Útočník se pomocí útoků snaží získat soukromá nebo neveřejná data, obsah komunikace, převzetí vlády nad počítačem nebo celou sítí či vyřadit server z činnosti. DoS (Denial of Service) je forma útoku cílená na internetovou službu (nejčastěji webovou stránku). DoS útok spočívá v tom, že jeden počítač (s jedním internetovým připojením) se snaží zahltit cílený server velkým množstvím požadavků. DDoS (Distributed Denial of Service) útok je prováděn z více počítačů (a více internetových připojení). Tedy rozdíl mezi DoS a DDoS je v počtu počítačů, které se podílí na útoku. [13]



Obrázek 2.2: Navázání spojení u TCP protokolu

Útočník se snaží vyčerpat zdroje serveru nebo infrastruktury, na které je útok cílen. Zdroji se myslí např. výpočetní výkon, zdroje operačního systému, šířka pásma sítě. Útočník posílá prostřednictvím počítačů, které se podílejí na útoku, velké množství požadavků. Daný server se snaží všechny tyto požadavky obsloužit. Pokud je nestíhá všechny zpracovávat, pro běžného (legitimního) uživatele se může zdát, že je pomalý nebo zcela nedostupný. DDoS útoky se dělí do tří kategorií:

Volumetrické DoS útoky – U volumetrických útoků útočník obvykle zahltí cílený server velkým množstvím paketů nebo množstvím otevřených spojení. Útok je úspěšný, když se vyčerpají zdroje daného serveru nebo síťové linky vedoucí k serveru. Distribuovaná varianta potom vypadá následovně. Protože je potřeba posílat velké množství paketů, útočník často napadne cizí zařízení, která jsou připojena k síti, pomocí škodlivého softwaru. Tato napadená zařízení, která se podílí na útoku, se označují jako zombie a síť takovýchto zařízení se označuje jako botnet.[5]

Amplifikační DDoS útoky – Tyto útoky využívají tzv. amplifikace (nebo-li zesílení) pro zvýšení síly útoku. Pro amplifikační útoky se používají se zneužívají síťové služby umožňující zesílení dopadu útoku. Amplifikace spočívá ve využití možnosti získání několikanásobné odpovědi na dotaz u něhož je odpověď zaslána na podvrženou (angl. spoofed) adresu (adresu oběti). Pro toto zesílení se zneužívají veřejně dostupné UDP služby, jako jsou UDP, NTP nebo například SNMP. [6]

Protokolové DDoS útoky – Tyto útoky zneužívají implementačních nebo návrhových chyb a nedostatků protokolů, které jsou na serveru oběti. Útočník se pak snaží prostřednictvím těchto chyb (nedostatků) vyčerpat zdroje na cílovém stroji (např. SYN Flood útok v kapitole 2.3).[5]

2.3 Útoky zneužívající TCP

ACK & PUSH ACK Flood – Během aktivního spojení, pakety s příznakem ACK nebo PUSH ACK nesou informace mezi klientem a serverem. Během tohoto útoku je na cílený server posíláno velké množství podvržených (spoofed) paketů s příznakem ACK se snahou tento server ochromit. Protože tyto pakety nepatří k žádnému aktivnímu spojení (z pohledu serveru), server spotřebuje více svých zdrojů na zpracování těchto paketů.

SYN-ACK Flood – Tento útok je obvykle prováděn menšími botnety, které posílají podvržené pakety s příznakem SYN na velké množství serverů a proxy serverů, které umožňují generovat velké množství paketů s příznaky SYN+ACK jako odpovědi na příchozí pakety s příznakem SYN. Tyto pakety nejsou směrovány zpátky na botnety, ale jsou směrovány do sítě oběti a často zahltí firwally oběti.

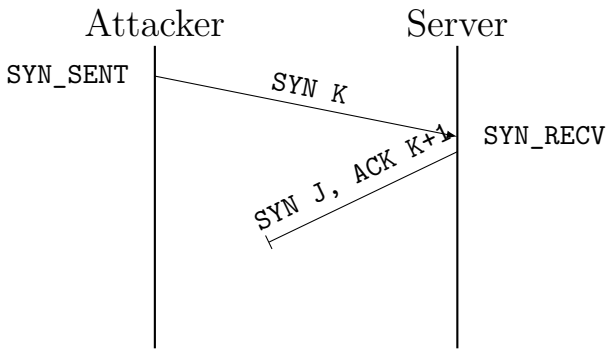
SYN Flood – Tento útok zneužívá třícestného navázání spojení tzv. three-way handshake (viz kapitola 2.1) , který slouží k navazování spojení protokolem TCP. Tomuto útoku se věnuje tato práce.

Server si při přijetí SYN paketu a odeslání SYN-ACK paketu vytvoří záznam o částečně vytvořeném spojení. Tento záznam se označuje TCB (Transmission Control Block) a je uložen v operační paměti serveru. Záznam TCB může mít i velikost přes 1300 B. Když by útočník odeslal velké množství SYN paketů, tak by se mohla vyčerpat operační pamět. Z tohoto důvodu se TCB záznamy ukládají do tzv. "backlogu", který se vytváří pro každý otevřený port.

Do backlogu lze uložit omezené množství TCB záznamů a toto číslo nebývá obvykle velké (např. pouze 256 záznamů). Operační systém udržuje TCB záznamy v backlogu do doby, než obdrží ACK paket, který bude přiřazen k danému TCB. Poté je spojení navázáno a operační systém odstraní TCB z backlogu a přesune si ho jinam do paměti. Záznam v backlogu může být pouze určitou dobu (např. 3 min), pokud během této doby nepřijde serveru ACK paket od klienta, záznam se z backlogu odstraní. Doba, po kterou je TCB uchovám v backlogu, je relativné velká, takže k vyčerpání kapacity backlogu dojde poměrně jednoduše.

Jakmile je kapacita backlogu vyčerpána, tak server nepřijímá žádné další žádosti o vytvoření spojení a daný port na serveru se stane nedostupný. Tato skutečnost je cílem SYN flood útoků, vyčerpat kapacitu backlogu a udržovat ji vyčerpanou. Tento protokolový DoS útok je snadnější na provedení než volumetrický DDoS útok, protože se posílá relativně malé množství TCP paketů.

Samotný útok probíhá tak, že útočník posílá SYN pakety, často s podvrženou (spoofed) IP adresou. Server se zachová přesně podle třícestného navázání spojení, tedy si vytvoří TCB záznam v backlogu a pošle SYN-ACK paket. Pokud paket dojde na podvrženou IP adresu, tak zařízení neočekává žádný SYN-ACK paket a zašle nazpátek paket s příznakem RST. Server tento paket příjme a smaže TCB z backlogu. Pokud, ale útočník zamezí odeslání RST paketu, záznam TCB v backlogu zůstane. Tím se poměrně rychle vyčerpá kapacita backlogu. Diagram SYN Flood útoku je na obrázku 2.3. [7]



Obrázek 2.3: Diagram SYN Flood útoku.

Zvýšení kapacity backlogu – První obranou proti SYN flood útokům může být zvýšení kapacity backlogu na koncové stanici. Ovšem v některých operačních systémech se zvyšování kapacity backlogu projevuje negativně. Vyhledávací algoritmy nejsou implementovány k prohledávání velkých backlogů. Útočníkovi stačí pouze zvýšit množství zasílaných paketů.[7]

Snížení časovače SYN-RECEIVED – Další lehce implementovatelnou metodou je zkrácení doby čekání mezi přijetím paketu s příznakem SYN a smazáním vytvořeného záznamu TCB. Tato metoda umožňuje rychlejší uvolnění prostředku pro legitimní provoz. Může ale také nastat situace, kdy nedojde k plně navázanému spojení legitimního spojení. Tato metoda se používá při překročení určitého prahu počtu obsazených TCB záznamů v backlogu.

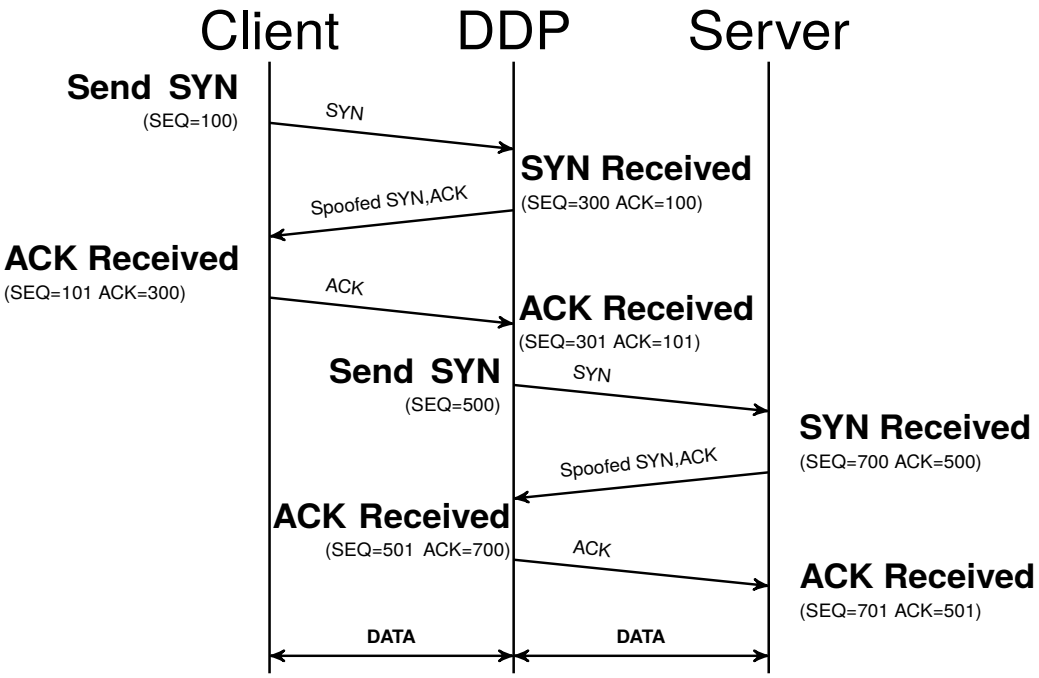
SYN Cache – Tato metoda spočívá v tom, že se při přijetí SYN paketu nevytváří celý záznam TCB. Alokace celého TCB záznamu je zpožděna do doby než bude navázáno úplné spojení. Informace o částečně navázaném spojení jsou ukládány do hashovací tabulky. Při příchodu SYN paketu jsou vybrány bity, které nejsou útočníkovi známé, které se zahashují spolu s informacemi do hashovací tabulky.

Tuto metodu je lepší použít na koncových stanicích, než na aktivním síťovém prvku, přes který prochází všechen provoz sítě. Aktivní prvek by musel mít funkci proxy serveru. Tedy by klient nejprve navázal spojení s proxy serverem, poté by proxy server navázal spojení se zařízením ve vnitřní síti. Následně by se tak museli i posílat datové TCP pakety, proxy server by musel přepočítávat sekvenční čísla, IP adresy, atd.[7]

SYN Cookies – Tato metoda rozšiřuje myšlenku přístupu SYN Cache. Server při přijetí SYN paketu nevytváří žádný záznam. Místo toho uloží všechny potřebné informace do sekvenčního čísla paketu SYN-ACK. Následně, pokud SYN paket patří do legitimního provozu, server přijme ACK paket od klienta. Informace, které byly uloženy do SYN-ACK paketu, lze získat z přijatého ACK paketu z čísla, které se označuje jako potvrzený bajt (acknowledgement number). Nevýhodou této metody je, že do 32 b sekvenčního čísla lze uložit pouze základní informace pro navázání spojení, takže TCP Options nelze do sekvenčního čísla uložit. V Options se například nachází MSS (Maximum Segment Size) pro nastavení maximální velikosti dat v jednom paketu.[7]

SYN Cookies metoda funguje tak, že přijme SYN paket od klienta. Následně proxy server vypočítá počáteční sekvenční číslo (ISN), do kterého jsou zahrnuty základní informace o klientovy na navazujícím spojení. Klientovy pošle SYN-ACK s ISN, legitimní klient pošle nazpátek ACK paket. Nelegitimní klient neodešle žádný paket, proxy server si nevytvořil žádný TCB záznam, takže se backlog nezaplní. Jakmile je navázáno spojení klient – proxy server, následuje navázání spojení proxy server - cílový server.

Nevýhoda této metody je stejná jako u metody SYN Cache. V obrázku lze vidět, že datové pakety se musí nejdříve poslat proxy serveru a poté je musí proxy server poslat cílovému serveru. Tedy se musí přepočítávat sekvenční čísla, IP adresy a kontrolní součty v IP a TCP hlavičce. Další nevýhodou je poměrně velká výpočetní náročnost, protože se musí sekvenční číslo (obvykle 2x) hashovat.[7]



Obrázek 2.4: SYN Cookies

Následující metody jsou implementovány v zařízení pro mitigaci útoků (viz kapitola 2.4):

SYN Drop modul (resp. metoda) slouží pro mitigaci SYN Flood útoků na základě rozhodovací tabulky 2.1. Vstupem modulu jsou pakety s příznaky SYN nebo ACK. Modul si udržuje čítače viděných paketů s příznakem SYN a ACK pro jednotlivé zdrojové IP adresy, které komunikují se sítí, kterou čistička provozu chrání. Čítače jsou pro určitých časových intervalech resetovány.

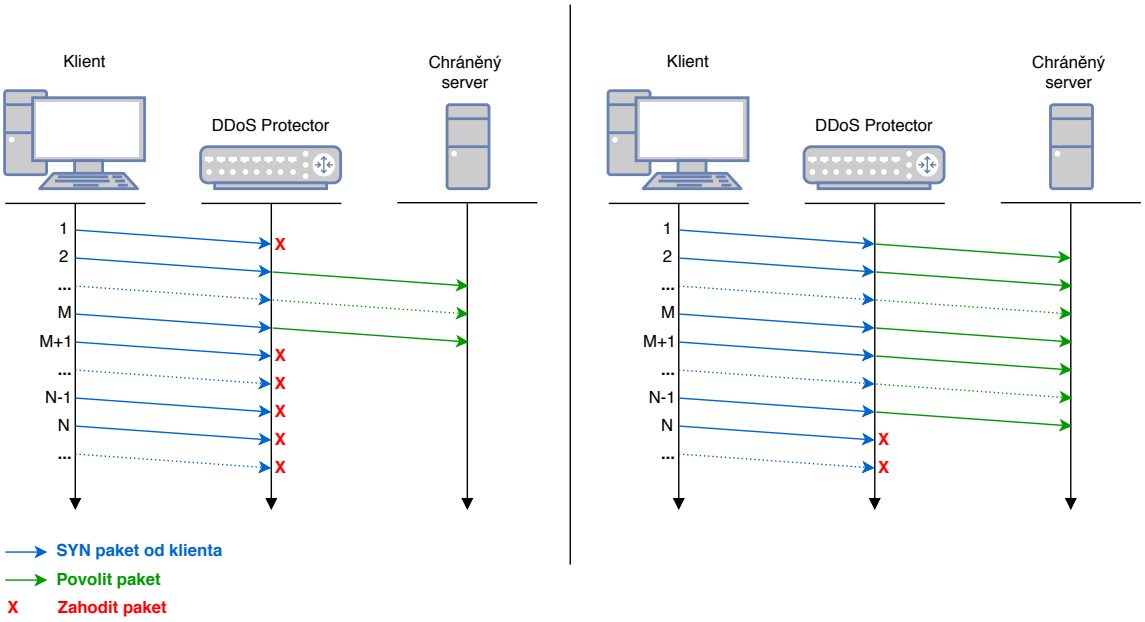
Modul rozhoduje pouze o zahození [v tabulce D] / přeposílání [v tabulce A] paketů s příznakem SYN, které jsou potřebné pro ustavení komunikace. Pakety s příznakem ACJ jsou přeposlány vždy. Pro určení operace, která se má provést s daným SYN paketem, slouží rozhodovací tabulka, ve které se vyhledá na základě aktuálních hodnot čítačů pro danou zdrojovou IP adresu příslušná operace. Tabulka obsahuje hodnoty M (relativně nízká mez) a N (relativně vysoká mez), které jsou konfigurovatelné a slouží k úpravě algoritmu.

		SYN			
	Rozsah	<1;1>	<2;M>	(M;N)	<N;∞)
ACK	<0;0>	D	A	D	D
	<1;∞)	A	A	A	D

Tabulka 2.1: Rozhodovací tabulka SYN Drop modulu.

První varianta zobrazuje situaci, kdy nebyl obdržen žádný paket s příznakem ACK (ACK = 0). Ve druhé variantě je znázorněno přeposílání SYN paketů při obdržení jednoho a více paketů s příznakem ACK (ACK >= 1)

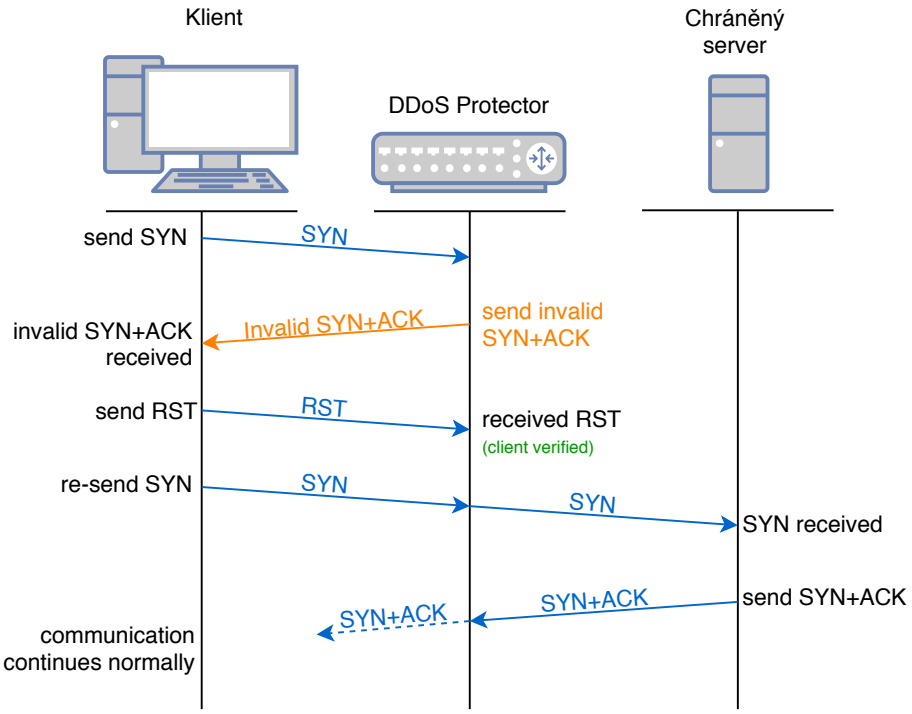
RST Cookies metoda – je založena na tom, že legitimní host by měl zaslat TCP paket s příznakem RST, pokud je TCP spojení v nesynchronizovaném stavu a je přijat pa-



Obrázek 2.5: SYN Drop - přeposílání paketů s příznakem SYN.

ket, který potvrzuje něco, co nebylo odesláno. Na základě tohoto vytváří submodul seznam ověřených hostů tzv. whitelist. Pro ověření hosta zahodí modul první SYN paket od tohoto hosta a pošle mu zpět neplatný paket s příznaky SYN+ACK. Legitimní host na tento paket odpoví paketem s příznakem RST, čímž je dokončeno jeho ověření. Následně podle standardu se klient znovu pokouší o spojení, tedy metoda nijak neovlivňuje chování klientských aplikací, protože vše se vyřeší na úrovni operačního systému.

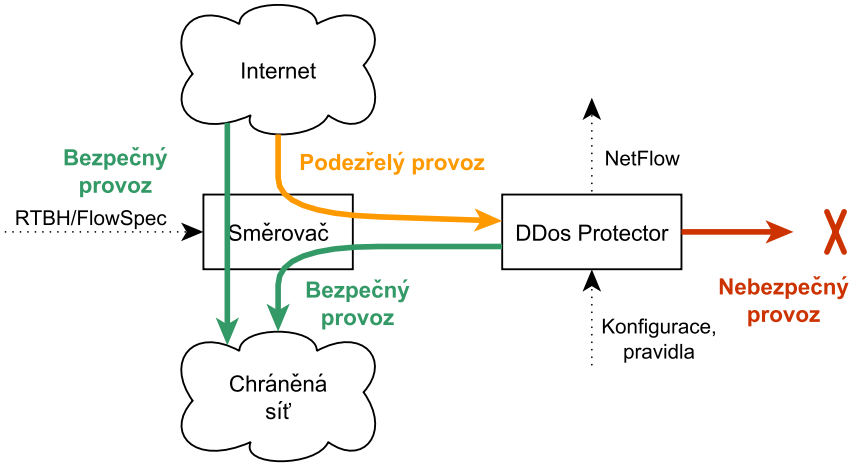
Nevýhodou této metody je to, že první pokus každého klienta o ustavení TCP spojení vždy selže, což by ale neměl být zásadní problém. Proces ověření uživatele je znázorněn na obrázku 2.6.



Obrázek 2.6: RST Cookies – ověření uživatele.

2.4 Zařízení pro ochranu před útoky

Zařízení pro ochranu před DDoS útoky tzv. DDoS Protector je aktivní síťové zařízení. Jeho úkolem je zahodit příchozí DDoS provoz z Internetu a propustit pouze provoz patřící legitimním uživatelům. DDoS Protector může být do sítě nasazen spolu se směrovačem. Směrovač přeposílá podezřelý síťový provoz do DDoS Protectoru. Ten zkontroluje každý přijatý paket a podle pravidel určí, jestli přijatý paket zahodí nebo přepošle zpět do směrovače. Poté je úkolem směrovače poslat „vyčištěný“ (legitimní) provoz do chráněné sítě. Informace o všech zahozených paketech a provozech jsou uloženy do logovacích souborů. Typické nasazení DDoS Protectoru je ukázáno na obrázku 2.7. [8]



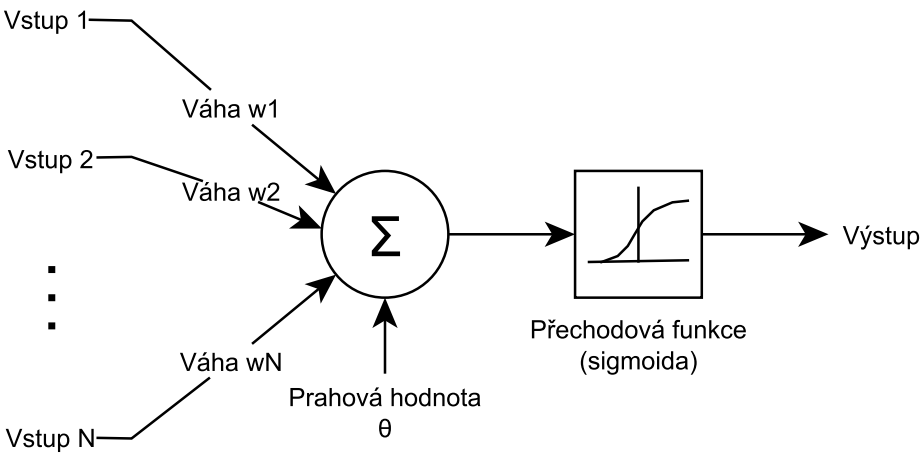
Obrázek 2.7: Typické nasazení DDoS Protectoru

Zařízení DDoS Protector je založeno na COMBO síťové kartě [9]. Rodina COMBO karet je sada desek s hardwarovou akcelerací zaměřených na zpracování dat v síti. Srdcem každé karty je programovatelné hradlové pole tzv. FPGA, dále externí paměti, PCI-Express a konektory síťového rozhraní.[8]

2.5 Neuronové sítě

V posledních letech vzrostl velký zájem o tzv. (umělé) neuronové sítě, což jsou matematické modely nervových systémů živých organismů. Lze je popsat i jako programovací paradigma, které se inspirovalo v biologii a umožňuje počítači „učit se“ z pozorovaných dat. U konvenčního přístupu programování počítači přesně říkáme, co má provést. Přesným opakem jsou právě neuronové sítě, kdy počítači neříkáme, jak má daný problém vyřešit (vypočítat). Počítač se učí z pozorovaných dat a snaží se sám nalézt řešení našeho problému. [2] [12]

Základním prvkem neuronových sítí je umělý neuron tzv. perceptron. V roce 1958 perceptron vytvořil Frank Rosenblatt. Je to vlastně matematický model biologického neuronu a jedná se o ústřední prvek neuronové sítě. Perceptron je inspirován neurony, což je základní prvek nervové soustavy živých organismů. Každý biologický neuron se skládá z těla tzv. somy, ke kterému jsou připojeny vstupy tzv. dentrity a jeden výstup tzv. axon. Rozhraní mezi dentritem jedno a axonem druhého neuronu se nazývá synapsí, jejíž váha představuje vliv jednoho neuronu na druhý neuron. [2]

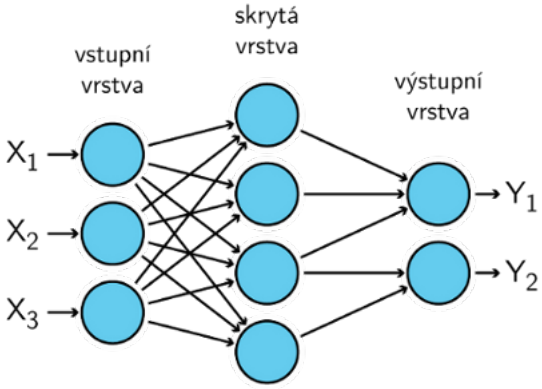


Obrázek 2.8: Perceptron

Perceptron je stejně jako biologický neuron tvořen několika vstupy, které vycházejí z povahy úlohy, a jeden výstup. Vstupy lze tedy chápat jako podněty z vnějšího okolí nebo jako výstup z jiných perceptronů (u vícevrstvé sítě). Každý vstup je rozšířen o váhu tohoto vstupu. Vlastní perceptron navíc také obsahuje prahovou hodnotu (v obrázku označeno jako Θ) tzv. potenciál neuronu, při jehož překonání je perceptron nabuzen a indikuje na svém výstupu signál ve formě aktivační funkce (též přenosová funkce). V neuronových sítích se používají různé aktivační funkce např. skoková přenosová funkce, sigmoidální přenosová funkce, přenosová funkce hyperbolické tangenty, atd. Výběr vhodné aktivační funkce má vliv na konvergenci procesu učení neuronové sítě. Výběr je mnohdy ovlivněn zkušenostmi z řešení obdobných úloh a mnohdy je nutno i experimentovat. [2]

Vícevrstvá neuronová síť

Vícevrstvá neuronová síť (MLP – Multi Layer Perceptron) se skládá ze vstupní vrstvy, výstupní vrstvy. Mezi těmito vrstvami se nachází skrytá vrstva (skryté vrstvy). Vícevrstvé neuronové sítě umožňují řešit složitější úlohy. Celá síť je tvořena opakováním základního prvku-perceptronu. Počet vstupů je dán počtem vstupů matematického modelu. Množství prvků ve vstupní a výstupní vrstvě je většinou dáno povahou úlohy. Množství skrytých vrstev a počet prvků se většinou volí jako maximum ze vstupní a výstupní vrstvy. Není možné shrnout vytváření skrytých vrstev do několika pravidel. Výzkumníci zkoumající neuronové sítě vytvořili mnoho heuristik pro skryté vrstvy. Jednotlivé heuristiky umožňují řešit různé problémy. Některé heuristiky (typy neuronových sítí) jsou popsány v následující části (viz podkapitola Typy neuronových sítí).



Obrázek 2.9: Vícevrstvá neuronová síť

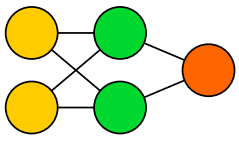
Vytvořenou neuronovou síť je potřeba naučit, aby mohla řešit problémy, na které jsme ji vytvořili. Rozlišuje se:

- Učení s učitelem – při učení s učitelem je neuronové síti předložen vzor. Na základě aktuálního nastavení je zjištěn aktuální výsledek. Ten poté porovnáme s očekávaným (vyžadovaným) výsledkem a určíme chybu. Následně se spočítá nutná korekce a upraví se hodnoty vah a prahů, aby se snížila hodnota chyby. Tento proces se opakuje dokud není dosaženo požadovaného výsledku.
- Učení bez učitele – při učení bez učitele nevyhodnocujeme výstup. Výstup nám není znám. Síť si sama třídí vstupy do skupin podle podobnosti a charakteristických znaků.

Typy neuronových sítí

V této podkapitole jsou uvedeny nejznámější a nejčastěji využívané typy neuronových sítí.

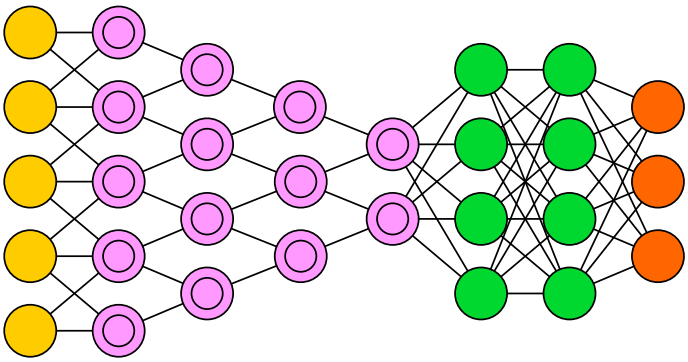
Feed Forward neural networks (FFNN) – Tyto neuronové sítě fungují tak, že výstup jedné vrstvy se použije jako vstup další vrstvy. Tyto sítě předávají informace vždy dopředu (angl. feed forward), nikdy zpět. V těchto sítích nevznikají žádné smyčky. Příklad této sítě lze vidět na obrázku 2.10, kde žluté kolečko značí vstupní prvek, zelené kolečko značí skrytý prvek a oranžové kolečko značí výstupní prvek. [16]



Obrázek 2.10: Feed Forward Neural network

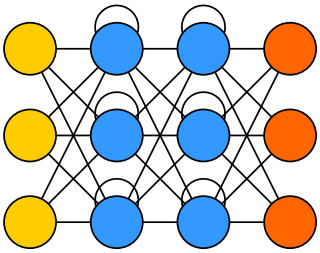
Konvoluční neuronové sítě – Konvoluční neuronové sítě se liší od základních neuronových sítí. Jejich hlavní použití je u zpracování vizuálních dat (obrázků), ale mohou také být použity například pro zpracování audio dat (hudby). Síť se skládá z tzv. skeneru. Typicky nechceme zpracovávat celý obrázek naráz, abychom neměli vrstvu

s velikým množstvím prvků (např. pro obrázek 200 x 200 pixelů by to byla vrstva s 40000 prvky). Obrázek se zpracovává postupně (např. 20 x 20 pixelů a po zpracování těchto pixelů se skener posune o jeden pixel doprava). Následují konvoluční vrstvy, které zpracují informace ze vstupní vrstvy. Příklad konvoluční neuronové sítě je na obrázku 2.11. [16]



Obrázek 2.11: Konvoluční neuronová síť

Rekurentní neuronové sítě – Jsou to FFNN s časovým posunem, tedy nejsou bezstavové. Mají spojení mezi jednotlivými průchody sítí (spojení v čase). Neurony získávají informacemi nejen z předchozí vrstvy, ale obsahují i zpětnou vazbu. Tedy výstup sítě nezávisí pouze na vstupu, ale i na vnitřním stavu, který je dán předchozími vstupy. To tedy znamená, že záleží na pořadí jednotlivých průchodů sítí. Tyto sítě se používají v oblastech, kde lze data reprezentovat jako sekvence a nemají časovou osu (např. obrázky nebo text). Obecně rekurentní neuronové sítě jsou dobrou volbou pro automatické doplňování informací. Příklad rekurentní neuronové sítě je na obrázku 2.12. [16]

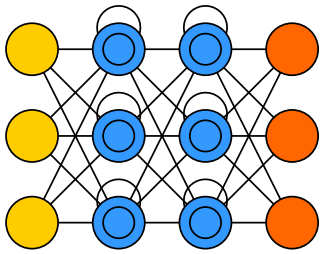


Obrázek 2.12: Rekurentní neuronová síť

Long / short term memory (LSTM) – Je varianta rekurentních sítí, které umožňují řešit úlohy s dlouhodobými závislostmi, kde každý neuron obsahuje paměťovou buňku a tři brány: vstupní, výstupní a forget bránu. Funkce těchto bran určují, jestli informace projde buňkou nebo bude „zastavena“.

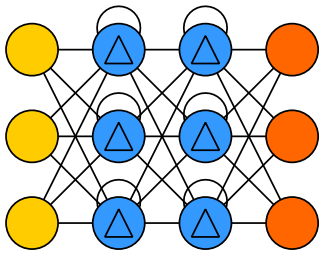
Vstupní brána určuje jaké množství informací z předchozí vrstvy bude uloženo v buňce. Výstupní brána určuje analogii ke vstupní bráně. Určuje jaké množství informace o stavu dané buňky bude odesláno následující vrstvě. Brána "na zapomenutí"určuje, jak už název napovídá, jaké množství informace bude zapomenuto (zahozeno).

LSTM neuronové sítě jsou schopny se naučit komplexní sekvence, jako je například kompozice jednoduché hudby. Každá brána udává váhu pro neuron v předchozí vrstvě, proto typicky spotřebují více prostředků. Příklad LSTM neuronové sítě je na obrázku 2.13. [16]



Obrázek 2.13: LSTM neuronová síť

Gated recurrent units (GRU) – jsou jednodušší LSTM neuronové sítě. Oproti LSTM sítím mají o jednu bránu méně a jsou propojeny trochu odlišně. Místo vstupní, výstupní a "zapomínací"brány mají obnovovací a resetovací bránu. Obnovovací brána určuje jaké množství informace bude ponecháno z posledního stavu i jaké množství informace bude použito z předchozí vrstvy. Resetovací brána je podobná "zapomínací"bráně v LSTM síti, ale je umístěna trochu odlišně. Ve většině případů GRU neuronové sítě fungují velice podobně jako LSTM neuronové sítě. Největší rozdíl mezi nimi je, že GRU sítě jsou rychlejší a jednodušší na běh, ale také i méně expresivní. Schéma GRU neuronové sítě je na obrázku 2.14. [16]



Obrázek 2.14: GRU neuronová síť

2.6 Využití neuronových sítí pro mitigaci DDoS útoků

Tento článek [19] se zaměřuje na využití neuronových sítí na detekci DDoS útoků. Zaměřuje se hlavně na srovnání čtyř typů neuronových sítí s metodami, které používají strojové učení. V článku uvádí, že metody se strojovým učením potřebují znalost sítě a statistické prvky. Použité neuronové sítě byly: Konvoluční neuronové sítě (CNN), Rekurentní neuronové sítě (RNN), Long Short-Term Memory Neural Network (LSTM) a Gated Recurent Unit Neural Network (GRU). Tyto sítě se ukázaly být úspěšné při zpracování velkých datových sad. Navíc sítě LSTM a GRU mají výhodu, že dokáží detekovat spojitosti mezi jednotlivými pakety, ať už legitimního síťového provozu, tak provozu, který je součástí útoku.

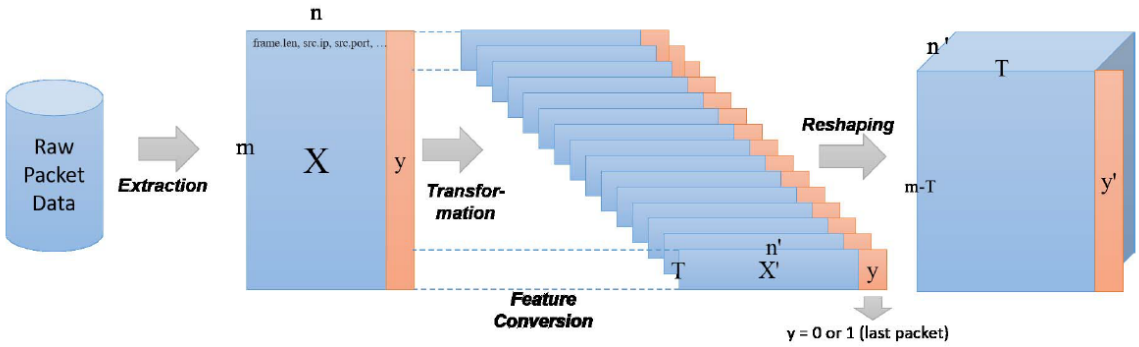
K experimentům s neuronovými sítěmi byla použita datová sada *UNB ISCX Intrusion Detection Evaluarion 2012 DataSet* (dále jen ISCX2012). Z datové sady se pro experimenty použil dvoudenní síťový provoz. Z tohoto provozu nejprve musely být extrahovány položky, které sloužily jako trénovací parametry uvedených neuronových sítí. Tabulka 2.2 obsahuje položky (parametry), které byly využity. V článku uvádí, že oproti metodám využívají strojové učení, nevyužívají statistické položky. Toto však není pravda, protože položky *tcp.analysis.ack_rtt*, *tcp.analysis.bytes_in_flight*, *tcp.analysis.duplicate_ack_num* patří mezi statistické položky.

Položka	Příklad položky	Typ položky
frame.encap_type	1	Binární
frame.len	805	Číselný
frame.protocols	eth:ip:tcp:http:data: data:data: data-text-lines	Textový
http.time	0	Číselný
icmp.length	203	Číselný
icmp.type	3	Binární
rc.request.command	PONG	Textový
irc.response.command	462,JOIN,353,366	Textový
tcp.ack	2.692e+03	Číselný
tcp.analysis.ack_rtt	0	Číselný
tcp.analysis.bytes_in_flight	1.460e+03	Číselný
cp.analysis.duplicate_ack_num	1	Číselný
tcp.dstport	2090	Binární
tcp.flags.urg	0	Binární
tcp.len	751	Číselný
tcp.srcport	80	Binární
tcp.window_size	12864	Číselný
udp.dstport	47666	Binární
udp.length	97	Číselný
udp.srcport	47521	Binární

Tabulka 2.2: Příklad 20 položek použitých jako parametry pro neuronové sítě.

Následující odstavec popisuje obrázek 2.15. Položky v tabulce 2.2 se dělí na tři typy: textové, číselné a binární. Počet položek je v obrázku uvedeno jako n. Tyto položky byly dále převedeny na vhodnější reprezentaci. Binární položky (např. TCP/UDP) byly převedeny

na seznam bitových čísel (v případě TCP/UDP na seznam o 16 bitech). Po transformaci vznikla matice o rozměrech $m \times n'$, kde m značí počet paketů, které se budou zpracovávat, a n' značí počet položek po transformaci. Také je navíc využito posuvné okno o velikosti T, které zajišťuje, že neuronové sítě objeví vzory mezi pakety, ať už z krátkodobého, tak i z dlouhodobého hlediska. Vzniknou tedy časová okna o velikosti T. Značka y, která se objevuje v každém okně, vyjadřuje, jestli se jedná o poslední paket.



Obrázek 2.15: Získání položek a jejich následná transformace. Obrázek byl převzat z [19].

V článku je použit návrh Obousměrné rekurentní neuronové sítě. V každém směru jsou použity dvě sequence-to-sequence rekurentní neuronové sítě. Tyto sítě umožňují sledovat předchozích paketů. V obou směrech byly využity buď sítě LSTM nebo sítě GRU (viz kapitola 2.5). V každé buňce bylo použito 64 neuronů a byla použita nelineární aktivační funkce *tanh*. V experimentech byly použity následující velikosti okna: 1, 5, 10, 50 a 100.

ISCX2012 obsahuje zachycený sedmidenní provoz, kde každý den je uložen do samostatného souboru. DDoS útok se objevil ve dvou dnech a jejich odpovídající soubory mají názvy: *Data14* a *Data15*. ISCX2012 obsahuje také záznamy o jednotlivých tocích, u kterých je také informace o tom, jestli daný tok patří do legitimního provozu nebo patří do útoku. V obou souborech je menší množství paketů patřících do útoků než paketů patřících do legitimního provozu. Pro učení neuronových sítí musí být poměr mezi útočícími pakety a legitimními pakety stejný. Aby byl tento poměr zachován, byly extrahovány všechny útočící pakety a k nim náhodně vybrány legitimní pakety.

Z experimentů, které jsou uvedeny v článku vyplývá, že všechny čtyři typy neuronových sítí dosahují přibližně stejných výsledků. LSTM model dosahuje o trochu lepší přesnosti. Experimentální výsledky ukazují, že sítě LSTM a GRU mají lepší výsledky oproti metodám využívající strojové učení.

Kapitola 3

Návrh

Tato kapitola je rozdělena na požadavky, které souvisí jak s konvenčním algoritmickým přístupem (ACK Spoofing), tak s neuronovými sítěmi. Následuje kapitola blíže popisující konvenční algoritmický přístup ACK Spoofing. V poslední části této kapitoly je popsán návrh neuronové sítě společně se vstupy a výstupy této sítě.

3.1 Požadavky

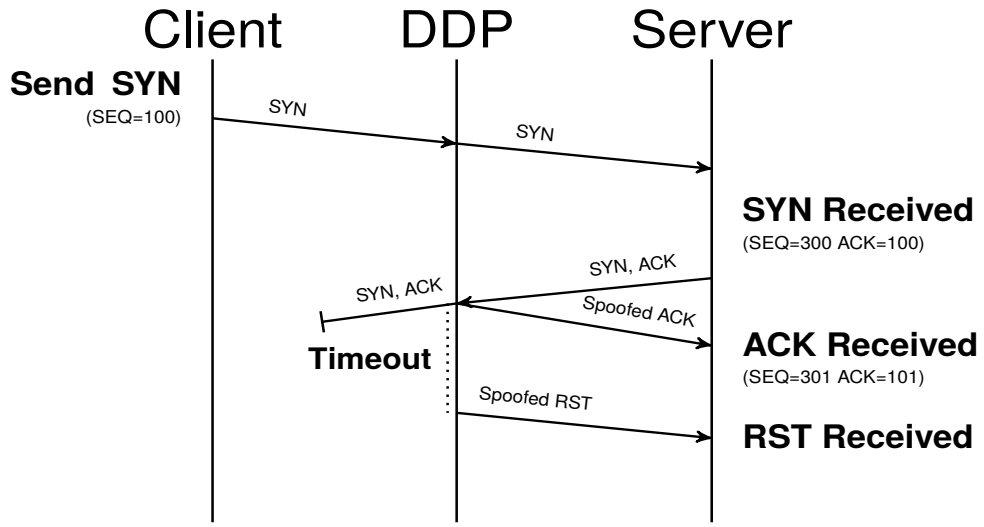
Při výběru metody, která by mohla být použita proti útokům, je potřeba vzít v úvahu několik požadavků. Vzhledem k tomu, že metoda bude použita v DDoS Protectoru, který bude nasazen na vysokorychlostních sítích, tak je potřeba, aby zasahoval do provozu, co nejméně. Tím se například myslí, aby metoda nepotřebovala přepisování hlaviček procházejících paketů, protože přepisování těch položek zvyšuje dobu zpracování paketu, tedy snižuje i propustnost. Z tohoto důvodu se nehodí metoda SYN Cookies (popsána v kapitole 2.3). Právě proto, že čistička bude použita na vysokorychlostních sítích, by nebylo z časového hlediska možné tuto metodu implementovat.

Další požadavkem na metodu je paměťová náročnost. I tento požadavek je důležitý, protože některé metody si potřebují uchovávat informace o procházejících paketech nebo tocích. A protože DDoS Protectorem může procházet několik milionů paketů za sekundu.

Také je potřeba se omezit pouze na metody, které budou detekovat útoky v reálném provozu, kdy nebudou k dispozici informace, které jsou známy při následné analýze na offline datech.

3.2 ACK Spoofing

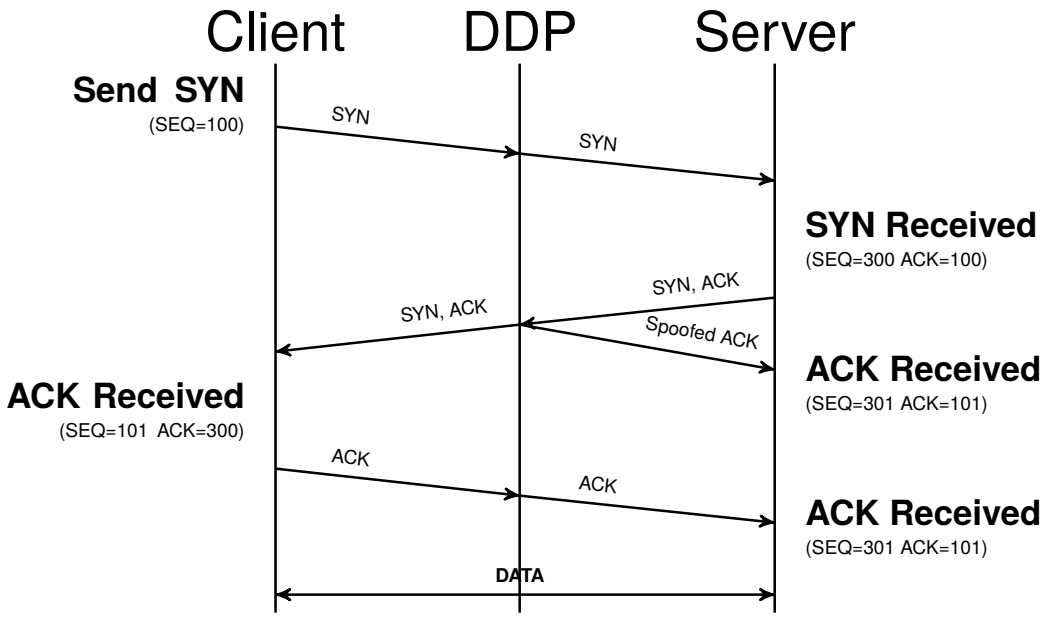
Tato metoda funguje tak, že proxy server přijme SYN paket od klienta, který ho přepošle cílovému serveru ve vnitřní síti. Server odpoví SYN-ACK paketem, který přijme proxy server, ten ho přepošle klientovi. Zároveň pošle serveru podvržený ACK paket, který cílový server očekává od klienta. Tím je na straně serveru spojení navázáno a záznam TCB se odstraní z backlogu. Proxy server si navíc uloží záznam, ve kterém jsou uloženy informace o spojení. Záznam je uložen do doby, než klient pošle skutečný (očekávaný) ACK paket, který se přepošle cílovému serveru. Poté se záznam z proxy serveru odstraní. [17]



Obrázek 3.1: ACK Spoofing - legitimní provoz.

Při legitimním provozu, kdy ale klient odešle RST paket, proxy server přepošle tento RST paket cílovému serveru. Zároveň smaže záznam o spojení. Plně navázané spojení na straně serveru se také ukončí.

Při nelegitimním provozu, kdy klient neodešle ACK ani RST paket zůstane záznam uložen v proxy serveru. Každý záznam je v paměti proxy serveru uložen pouze po omezenou dobu (tzv. timeout). Tato doba musí být menší než doba, po kterou jsou uloženy záznamy v backlogu na cílovém serveru. Jakmile tato doba vyprší, záznam bude odstraněn a cílovému serveru se pošle RST paket, který ukončí spojení na jeho straně.



Obrázek 3.2: ACK Spoofing - nelegitimní provoz.

Výhoda této metody spočívá v tom, že klient a cílový server komunikují s proxy serverem pouze při navazování spojení. Jakmile je spojení navázáno, mohou si klient a cílový server zasílat pakety přímo navzájem.

3.3 Návrh neuronové sítě

Je potřeba zdůraznit, že cílem práce je kromě konvenčního algoritmického přístupu ACK Spoofing možné detekovat SYN Flood útok, který se nachází v legitimním provozu. Zároveň je potřeba zjistit, jestli neuronová síť nebude označovat legitimní provoz jako útok, což ve větší míře není přijatelné. Předmětem práce není zjištění náročnosti neuronové sítě na paměť nebo výkon. Také neobsahuje dobu zpracování jednoho paketu.

Součástí návrhu neuronové sítě je i výběr a návrh vstupních položek pro tuto neuronovou síť. Mezi vstupní položky pro neuronovou síť jsem zařadil:

- Typ protokolu
- Velikost TCP hlavičky
- Z TCP:
  - Velikost okna
  - Příznaky
  - Zdrojový port
  - Cílový port

Mezi vstupní položky neuronové sítě jsem nepoužil zdrojovou ani cílovou IP adresu z IP hlavičky. Protože datové sady byly uměle vytvořeny, objevuje se v nich relativně malý počet IP adres, což by mohlo ovlivnit výsledky experimentů. Stejně tak jsem z TCP hlavičky nepoužil pořadové číslo (angl. sequence number) a potvrzovací číslo (angl. acknowledgment number). Je to z toho důvodu, protože pakety patřící do legitimního provozu mají tato čísla nastavená tak, jako u normálního provozu, avšak pakety patřící do útoku jsou nastavena na hodnotu 0. I když jsem pořadové ani potvrzovací číslo nepoužil jako vstup pro neuronovou síť, jsou součástí souboru s ostatními vstupními položkami pro neuronovou síť. To jsem udělal z důvodu, kdyby při provádění experimentů bylo potřeba z jakéhokoliv důvodu tyto dvě položky využít.

Položky: typ protokolu, zdrojový a cílový port a příznaky jsem také převedl na jednotlivé bity. To jsem udělal z toho důvodu, protože neuronová síť se lépe učí na číslech, které jsou v binární podobě než, když jsou v dekadické soustavě.

Výsledný soubor se skládá z následujících položek:

- Položky 0 až 15 reprezentují typ protokolu
- Položka 16 – pořadové číslo paketu
- Položka 17 – potvrzovací číslo paketu
- Položka 18 – velikost TCP hlavičky
- Položka 19 – velikost okna
- Položka 20 až 27 – příznaky

- Položka 28 až 43 – zdrojový port
- Položka 44 až 59 – cílový port
- Položka 60 – kategorie paketu (viz následující odstavec)

Součástí souboru se vstupními položkami je i označení jednotlivých paketů. Toto označení určuje třídu paketu, tedy jestli se jedná o paket patřící do třídy:

- SYN Flood útok
- legitimní provoz, ale má pouze příznak SYN
- zbylé pakety, tedy pakety mající např. příznaky ACK, SYN+ACK, FIN, atd.

Výstupem neuronové sítě je tedy zařazení klasifikovaného paketu do jedné z výše uvedených tříd. Z pohledu práce jsou nejdůležitější získané výsledky pro první dvě třídy, tedy pro SYN Flood útok a legitimní pakety se SYN příznakem. Pro co nejpřesnější naučení neuronové sítě je potřeba zachovat kontext mezi pakety. Kdyby vstupní data byla omezena pouze na pakety s příznakem SYN, které by patřily do útoku a legitimního provozu, nebyl by tento kontext zachován a pro neuronovou síť by bylo mnohem složitější tyto pakety od sebe odlišit.

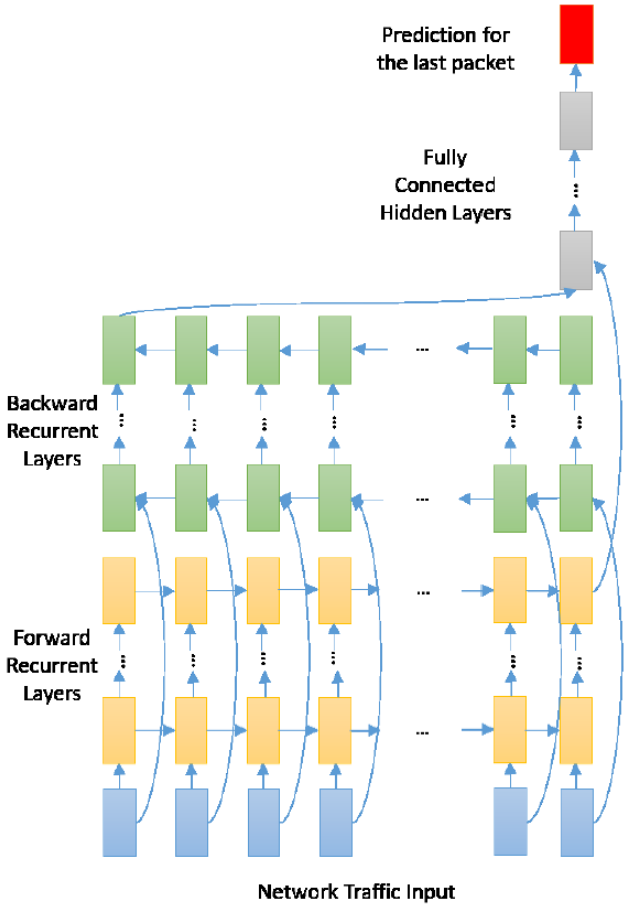
Pro experimenty jsem použil model neuronové sítě z článku [19]. Model je složen z dvousměrné rekurentní neuronové sítě (angl. Bidirectional Recurrent Neural Network). Směry se v modelu skládají z Forward Recurrent Layers a Backward Recurent Layers. V každé z těchto vrstev se nachází dvě vrstvy, které umí zpracovávat sekvence. Tyto vrstvy tedy umožňují při zpracování daného paketu, využít informace o paketech, které zpracovaly před ním. Neurony v těchto vrstvách jsou v experimentech (v kapitole 5.2) implementovány jako LSTM a GRU (viz kapitola 2.5). Obě tyto implementace obsahují paměť, díky které si pamatují předchozí pakety. Implementace GRU je pouze jednodušší varianta implementace LSTM. V základním nastavení sítě jsem použil (stejně jako v článku) 64 neuronů v každé vrstvě. U každého neuronu jsem použil nelineární aktivační funkci:

$$f(x) = tanh(x).$$

Následně se zřetězí výstupy obou vrstev (Forward a Backward) jsou spojeny do jedné vrstvy (označeny jako Fully Connected Layers). Zde byla použita Rectified Linear Unit (RELU) jako aktivační funkce. A ve výstupní vrstvě byla jako aktivační funkce použita Sigmoid funkce:

$$f(x) = \frac{1}{1 + e^{-x}}.$$

Před každými dvěmi rekurentními neuronovými vrstvami a před každou Fully Connected vrstvou byla použita Batch Normalization vrstva, která slouží k urychlení procesu učení neuronové sítě. Celkový model neuronové sítě je znázorněn na obrázku 3.3.



Obrázek 3.3: Získání položek a jejich následná transformace. Převzato z [19]

Kapitola 4

Implementace

Tato kapitola obsahuje implementační detaily. V první části je popsána prototypová implementace ACK Spoofingu. V následující části je popsáno rozšíření této implementace jako modulu. V další části je popsán program pro extrakci a úpravu dat do potřebného formátu pro neuronové sítě (viz kapitola 3.3). V poslední části je popsána implementace neuronové sítě.

4.1 Prototypová implementace

Program jsem implementoval v programovacím jazyce C. Pro příjem a zpracování paketů jsem použil knihovnu pcap.

Informace o vytvářeném spojení jsou v programu interpretovány jako záznam. Záznam je tvořen ze zdrojové a cílové IP adresy a portu. V záznamu je také uložena časová značka, která značí čas vytvoření záznamu a sekvenční číslo, které potvrzuje SYN+ACK pakety z cílového (chráněného) serveru. Časová značka se použije pro určení, jestli vypršela doba (angl. timeout), po kterou má záznam nechat uložený.

Pro uložení záznamu o spojení jsem použil strukturu typu fronta (FIFO), kterou jsem implementoval jako dynamický lineární seznam. Záznamy jsou ve frontě vytvářeny a seřazeny podle časové značky vytvoření záznamů. Z fronty lze záznamy odstraňovat standardně ze začátku. Navíc lze odstraňovat i záznamy, které jsou uložené i uprostřed.

Průběžně se kontroluje časová značka záznamu na začátku fronty. Při kontrole se k časové značce přičte hodnota maximální doby uložení (timeoutu). Pokud výsledek překračuje hodnotu aktuálního času, záznam se odstraní a serveru se odešle podvržený paket s příznakem RST pro ukončení spojení. Záznamy na začátku fronty se kontrolují, dokud součet je větší než aktuální čas.

Funkce pro odebrání záznamu, který je uložen uprostřed fronty, jsem implementoval z důvodu legitimního provozu. Záznam se vytvoří při přijetí SYN+ACK paketu od serveru. Pokud přijde paket s příznakem ACK dřív než vyprší timeout, musí se smazat záznam o spojení, do kterého patří tento paket. Tento záznam se však nenachází na začátku fronty, proto jsem přidal funkci pro odebrání záznamu, který se nemusí nacházet na začátku fronty. Přijatý ACK paket od klienta je následně odeslán serveru.

Pokud by bylo potřeba z fronty smazat úplně poslední záznam, bylo by nutné projít všechny předchozí záznamy. Vyhledání takového záznamu by bylo časově velmi náročné. Z tohoto důvodu je vedle fronty použita další struktura – hashovací tabulka. Díky hashovací tabulce bude možno rychleji vyhledat záznamy, které se nenachází na začátku fronty. Využil

jsem implementaci `fasthashtable`. Jako klíč se použije zdrojová a cílová IP adresa a port. V hashovací tabulce je potom uložena adresa do paměti, kde je uložen hledaný záznam. Není tedy potřeba sekvenčně procházet frontu (lineární seznam).

Program byl implementován jako celek, tedy implementace algoritmu je přímo propojená se zpracováním a odesláním paketů. Program byl také implementován pouze s podporou adres IPv4.

4.2 Implementace jako modul

ACK Spoofing program bude nejlepší naimplementovat jako modul. Funkčnost modulu bude spočívat ve volání jednotlivých funkcí, které budou volány aplikací, která bude fungovat nad modulem a budou provádět algoritmus ACK Spoofing. Modul byl naimplementován nad hardwarovým rozhraním SZE.

Nejprve jsem provedl úpravu alokace paměti. Fronta, která se používá k uchování informací o spojeních (záznamy) a kontrole vypršení časového limitu (angl. timeoutu), v prototypové implementaci alokovala paměť pokaždé, když bylo potřeba vytvořit záznam o spojení. To nastane tehdy, když program přijme SYN+ACK paket odeslaný ze serveru (chráněné sítě). Tento způsob fungování je samozřejmě časově náročný. Nyní se na začátku inicializace modulu alokuje celá paměť určená pro položky fronty. Při spuštění modulu je potřeba nastavit maximální velikost (maximální počet záznamů) fronty a hashovací tabulky. Hashovací tabulka je nejefektivnější, když je její velikost (počet řádků) mocninou čísla 2, proto je doporučeno i velikost fronty nastavit na tuto hodnotu. Modul umožňuje, aby mu aplikace běžící nad ním mohla tuto hodnotu předat.

Modul bude možné volat aplikací běžící nad tímto modulem ve více vláknech. Je nutné, aby aplikace běžící nad modulem, uměla posílat síťový provoz do jednotlivých vláken (instancí modulu), ve kterých probíhá daný provoz. Například: Server pošle SYN+ACK paket a vytvoří se záznam v určité instanci modulu (vláknu). Při další komunikaci ACK nebo RST paketu od klienta je potřeba tento paket poslat do instance modulu (vlákna), kde se uložil předchozí záznam.

Tato implementace oproti prototypové implementaci odděluje část pouze s funkcemi realizující algoritmus ACK Spoofing a část obsluhující přijímání a odesílání paketů. Dále podporuje, jak adresy IPv4, tak IPv6.

4.3 Implementace programu na úpravu dat pro neuronovou síť

Pro provedení experimentů s neuronovými sítěmi je potřeba mít data v určitém formátu (viz kapitola 3.3). Daty se zde myslí pakety, u kterých se bude během experimentů rozhodovat, jestli se jedná o paket patřící do jedné z tříd: SYN Flood útok, legitimní provoz se SYN příznakem a nebo zbylý legitimní provoz. Pro učení neuronových sítí je potřeba, aby počet paketů pouze s příznakem SYN patřící do útoku byl pokud možno stejný, jako počet paketů pouze s příznakem SYN patřící do legitimního provozu. Výsledný soubor, který bude vstupem neuronové sítě, obsahuje tři typy paketů. Jsou to pakety pouze s příznakem SYN patřící do útoku, pakety pouze s příznakem SYN patřící do legitimního provozu a zbývající pakety patřící do legitimního provozu.

Program funguje tak, že čte více souborů ve formátu csv [1]. Tyto soubory určují, jestli pakety v něm obsažené jsou součástí útoku nebo součástí legitimního provozu. V jednom souboru jsou tedy pakety patřící do útoku a v druhém jsou pakety patřící do legitimního provozu (ať už legitimního, jen s příznakem SYN, tak i zbylý provoz). U jednotlivých paketů je i časová značka jeho přijetí. Jednotlivé pakety jsou v souboru odděleny znakem konce řádku. Program počítá se skutečností, že časové značky jsou „zarovnané“ tak, aby pakety patřící do útoku byly z pohledu času mezi pakety patřící do legitimního provozu (viz kapitola 4.4.

Program postupně přečte poslední nepřečtený a nepoužitý paket z každého souboru a vybere ten, který má časovou značku s nejmenší hodnotou oproti ostatním. Může dojít k tomu, že daný soubor bude uzamčen pro další čtení, v tom případě se další pakety z tohoto souboru již číst nebudou. Při porovnávání časových značek, program pracuje i s nanosekundy, protože útok probíhá oproti legitimnímu provozu na relativně malém časovém intervalu.

4.4 Předpříprava dat pro program na extrakci a úpravu dat pro neuronovou síť

Soubory, které jsou použity jako vstup programu na extrakci a úpravu dat pro neuronovou síť je potřeba předpřipravit. Jako legitimní provoz jsem použil pakety z datové sady ISCX-IDS-2012 [4], která byla použita v článku [19]. Tato datová sada obsahuje zachycený provoz v období jednoho týdne a je rozdělen do souborů odpovídající jednotlivým dnům. Pro legitimní provoz jsem si vybral soubor `testbed-16jun.pcap`, který neobsahuje žádný DDoS útok, který by mohl ovlivnit výsledky neuronové sítě.

Pro útok jsem si zvolil data z datové sady ACS MILCOM 2016 Dataset E [14], ve kterém se nachází SYN Flood útok. Tento útok obsahuje přibližně 60000 paketů. Pakety patřící do tohoto útoku jsem vyextrahoval pomocí nástroje Wireshark.

V souboru s legitimním provozem je menší poměr paketů pouze s příznakem SYN vůči celkovému počtu paketů. Z tohoto důvodu jsem z legitimního provozu vybral jako do trénovacích dat provoz, který obsahoval 450000 paketů. Z nich přibližně 3600 bylo pouze s příznakem SYN. Z tohoto provozu jsem kvůli velkému množství vedlejšího provozu odstranil pakety, jejichž velikost přesahovala 200 Bytů. Odstranily pouze pakety, které neobsahovaly příznak SYN. Tím se zredukoval celkový počet paketů na polovinu. K tomuto provozu jsem z provozu patřícího do útoku vybral také přibližně 3600 paketů. Vznikly tedy dva soubory s trénovacími pakety pro neuronovou síť, kde první soubor obsahuje vedlejší legitimní provoz a pakety legitimního provozu pouze s příznakem SYN, druhý soubor obsahuje pouze pakety s příznakem SYN patřící do útoku.

Stejným způsobem jako jsem vytvořil soubory s trénovacími pakety, jsem vytvořil dva soubory s testovacími pakety. Rozdíl při extrakci těchto paketů je, že jsem nemohl vybrat stejný legitimní provoz, ani stejné pakety patřící do útoku. Proto jsem vybral provoz, který se nachází těsně za provozem, který jsem vybral jako trénovací data. Testovací data obsahují přibližně desetinu paketů s příznakem SYN oproti trénovacím datům.

Protože soubor s pakety patřící do útoku je vybrán z jiné datové sady než soubor s pakety legitimního provozu, mají i jiné časové značky. Pakety patřící do útoku mají časové značky v posunuté vůči paketům patřící do legitimního provozu. Pokud by se tyto soubory s těmito

[1] CSV soubor je jednoduchý souborový formát, který sestává z řádků, kde jsou jednotlivé položky odděleny znakem čárka.

různými časovými značkami spojily dohromady, došlo by k tomu, že by všechny pakety patřící do útoku byly za všemi pakety patřící do legitimního provozu. V mém případě jsem musel časové značky u paketů patřící do útoku posunout tak, aby se tyto pakety nacházely „uvnitř“ legitimního provozu. Tento posun jsem provedl pomocí nástroje `editcap`. Tento posun jsem musel provést pro, jak pro trénovací data, tak i pro testovací data.

Poté, co jsem vyextrahoval a časově posunul pakety, jsem převedl data v .pcap souborech na soubory typu csv. Tento převod jsem provedl pomocí nástroje `tshark`. Příkaz pro převedení `.pcap` souboru na CSV soubor:

```
tshark -r <nazev_soboru>.pcap -T fields -E separator=,
-e frame.time
-e frame.encap_type
-e frame.len
-e frame.protocols
-e ip.src
-e ip.dst
-e tcp.srcport
-e tcp.dstport
-e icmp.length
-e icmp.type
-e tcp.seq
-e tcp.ack
-e tcp.len
-e tcp.flags
-e tcp.window_size
```

4.5 Implementace neuronové sítě

Neuronovou síť jsem implementoval v programovací jazyce Python ve frameworku Keras [1]. Tento framework umožňuje jednoduché a rychlé prototypování neuronových sítí. Je schopen pracovat nad open source knihovnami TensorFlow, CNTK a Theano, které provádí numerické výpočty. Pro své experimenty jsem použil knihovnu TensorFlow.

Pro zvolené metriky: precision, recall a f1, které jsou popsány v kapitole 5.2, jsem využil knihovnu scikit-learn [3]. Tato knihovna napsaná v programovací jazyce Python je určená pro jednoduché a efektivní výpočty spojené s data minigem [2] a analýzou dat. Tuto knihovnu jsem využil, protože implementace použitých metrik byla ve frameworku Keras nepřesná, a proto byla odstraněna. Z knihovny scikit-learn jsem také použil funkci pro vykreslení tzv. confusion matrix, která ukazuje efektivnost klasifikace neuronové sítě (viz kapitola 5.2).

Při měření si získané hodnoty ukládám do polí, které odpovídají jednotlivým třídám a pro každou třídu se měří tři výše uvedené metriky. Hodnoty se měří na testovacích datech a po každé epoše [3].

[2] Data mining je analytická metodologie získávání netriviálních skrytých a potenciálně užitečných informací z dat.

[3] Epocha je jedno naučení neuronové sítě na trénovacích datech.

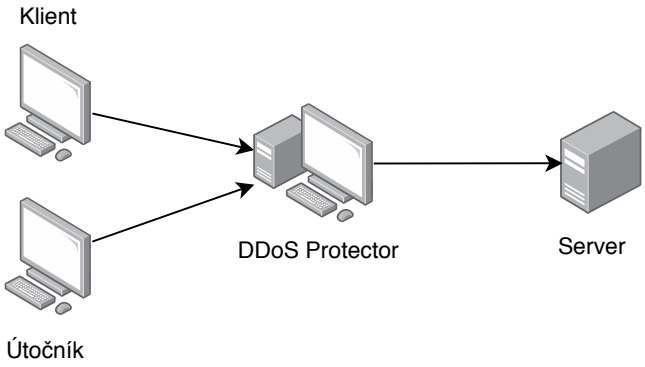
Kapitola 5

Získané výsledky

Tato kapitola obsahuje získané výsledky. V první části kapitoly jsou výsledky konvenčního algoritmického přístupu ACK Spoofingu. V druhé části jsou popsány výsledky experimentů nad neuronovou sítí.

5.1 Testování metody ACK Spoofing

Metodu ACK Spoofing jsem otestoval ve virtuálním prostředí. Testovací prostředí je složeno ze čtyř virtuálních zařízení. Jak lze vidět na obrázku 5.1. Uprostřed obrázku se nachází DDoS Protector, na kterém byl spuštěn ACK Spoofing. K němu jsou zleva připojeny zařízení, na kterém je klient a útočník. Zprava je k DDoS Protectoru připojen server, ke kterému se bude snažit klient připojit.



Obrázek 5.1: Testovací prostředí metody ACK Spoofing.

Klient se bude snažit navázat spojení ke službě FTP na portu 21, tedy se bude snažit přenést soubor. Útočník provede SYN Flood útok na tento port, aby zamezil klientovi navázat spojení. Útok by byl proveden pomocí nástroje **hping3**, který umožňuje odeslání paketů s určitým příznakem na jiný stroj na daný port (službu). Zdrojový port by byl nastaven na hodnotu, která neodpovídá žádnému právě využívanému portu na stroji útočníka. Útočník by měl podle třícestného navázání spojení (viz 2.2) od serveru přijmout paket s příznaky SYN+ACK a nazpět odeslat paket s příznakem RST, protože port není používaný žádnou aplikací. Z tohoto důvodu jsem do firewallu na stroji útočníka přidal pravidlo, které

zahodí odchozí pakety s příznakem RST. Serveru tedy nepřijde žádný paket s příznakem RST a spojení nebude ustanoveno ani ukončeno. Záznam představující toto spojení zůstane uloženo backlogu (viz 2.1). Otestování ACK Spoofingu jsem provedl následovně:

1. Pro následující dva testy bylo pravidlo o zákazu posílání paketů s příznakem RST odstraněno z firewallu. Jako první jsem provedl zkoušku, že lze úspěšně zahájit přenos souboru z klienta na sever, což bylo provedeno úspěšně.

2. Dále jsem provedl ověření, že metoda ACK Spoofing umožňuje i navázání legitimního spojení. Spustil jsem metodu ACK Spoofing. Poté jsem se pokusil o zahájení přenosu souboru, což bylo provedeno úspěšně. To ukazuje na to, že metoda přepošle odpověď od klienta na server a odstraní záznam spojený s tímto spojením ze svých struktur (timeout fronta a hashovací tabulka).

3. Pro následující testy bylo do firewallu přidáno pravidlo o zakázání posílání paketů s příznakem RST. Následně jsem ze stroje útočníka zahájil SYN Flood útok na server na port 21. ACK Spoofing byl během tohoto útoku vypnutý. Následně jsem se stejně jako v kroku 1 pokusil zahájit přenos souboru z klienta na server. To se však podle očekávání nepodařilo v důsledku probíhajícího SYN Flood útoku. Poté jsem tento útok ukončil.

4. V dalším pokusu jsem provedl ověření, že metoda ACK Spoofing funguje tak, jak je očekáváno. Před spuštěním SYN Flood útoku na server jsem spustil metodu ACK Spoofing. Poté jsem zahájil útok, který se skládal pouze z velmi malého množství paketů, které měly různé zdrojové porty, aby bylo jednodušší dohledat k nim přidružená spojení. Metoda ACK Spoofing skutečně při navazování spojení odeslala zpět na server podvržené pakety s příznaky ACK. Částečně vytvořená spojení byla ustanovena a záznamy v backlogu byly přesunuty do operační paměti. Zde byly uchovány době, která odpovídá času uchování záznamů v timeout frontě v metodě ACK Spoofing. Při sledování síťového provozu na serveru byly zaznamenány pakety s příznaky RST, které odpovídaly těmto ustanoveným spojením.

5. V tomto testu jsem pracoval se stejnou situací jako v kroku 2 s tím rozdílem, že jsem zapnul metodu ACK Spoofing. Následně jsem se při běžícím útoku a zapnuté metodě ACK Spoofing pokusil zahájit přenos souboru z klienta na server. Zahájení přenosu souboru proběhlo úspěšně a soubor se začal přenášet.

5.2 Experimenty s neuronovou sítí

U experimentů jsem jako vstupní data pro neuronovou síť použil datové sady ISCX2012 a ACS MILCOM 2016 E, které jsem popsal v kapitole 4.4. V této kapitole jsem taky popsal velikost a obsah trénovacích a testovacích dat, které byly následně zpracovány programem pro finální vstup pro neuronovou síť (viz kapitola 4.3).

Během experimentů musí neuronová síť pro každý paket určit, do jaké třídy patří. Jak již bylo řečeno v kapitole 3.3, třídy jsou tři: SYN Flood útok, legitimní provoz pouze se SYN příznakem a zbylý provoz. Z hlediska práce jsou nejdůležitější výsledky pro první dvě třídy. Třetí třída (zbylý provoz) je v síti použit pouze pro zachování kontextu mezi jednotlivými pakety.

U jednotlivých experimentů jsem použil metriky precision, recall a f1. Tyto metriky se vypočítají pomocí True Positives, True Negatives, False Positives a False Negatives,

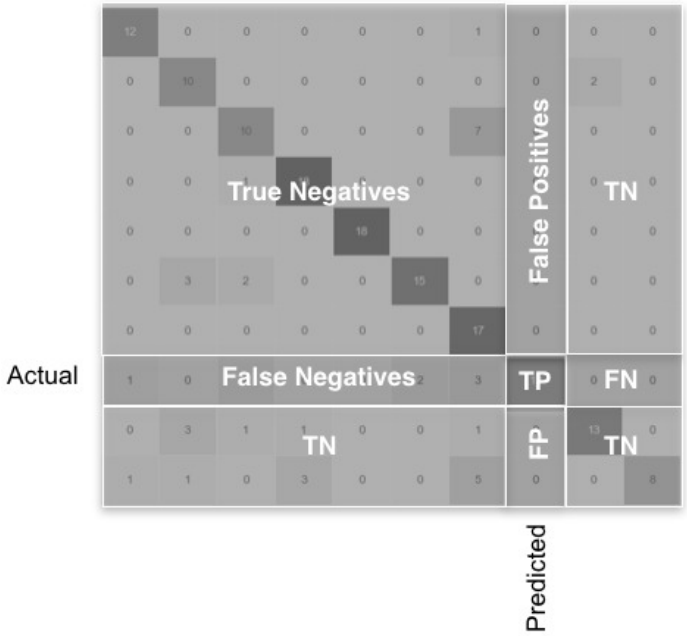
které jsou znázorněny na obrázku 5.2 znázorňující Confusion matrix, kde řádky odpovídají skutečným hodnotám (třídám) a sloupce odpovídají odhadnutým hodnotám (třídám) neuronové sítě.

True Positives jsou jednotlivé hodnoty v diagonále matice, což odpovídá shodě skutečné třídy a odhadnuté třídy nad daty. Je to tedy počet správných pozitivních detekcí.

True Negatives je součet všech hodnot matice vyjma sloupce a řádku matice, v jejichž průniku se nachází hodnota True Positive.

False Positives je součet všech hodnot ve sloupci, kromě hodnoty True Positives (počet chybných detekcí).

False Negatives je součet všech hodnot v řádku, kromě hodnoty True Positives. To odpovídá chybné detekci pro danou třídu.



Obrázek 5.2: Confusion matrix - vysvětlivky, převzato z [10]

Precision je možné nazvat míra přesnosti nebo kvalita. Definovat ji lze jako podíl počtu správně nalezených případů (v tomto případě paketů) a počet všech detekovaných řešení (celkový počet paketů pro odhadovanou třídu) podle algoritmu, jenž byl implementován (v této práci to je neuronová síť). Výsledek se pohybuje v rozmezí $[0; 1]$, přičemž 0 je nulová přesnost (kvalita) a 1 je nejvyšší přesnost (kvalita) pro danou třídu. Metrika precision lze vypočítat pomocí vzorce: [18], [15]

$$precision = \frac{true\,positive}{true\,positive + false\,positive}.$$

Recall je možné nazvat jako míra úplnosti nebo kvantita. Definovat ji lze jako podíl počtu správně nalezených případů (paketů) a všech případů (paketů) pro danou třídu.

Výsledek se také pohybuje v rozmezí $[0; 1]$, přičemž 0 je nulová kvantita a 1 je maximální kvantita pro danou třídu. Metrika recall lze vypočítat pomocí vzorce: [18], [15]

$$recall = \frac{true\,positive}{true\,positive + false\,negative}.$$

Metrika F1 (též F-measure) kombinuje výsledky výše uvedené metriky precision a recall. Rozsah výsledků této metriky je na intervalu $[0; 1]$, přičemž skóre (výsledek) blížící se 0 může vypovídat buď o nízkých hodnotách precision nebo recall. Také může vypovídat o velmi nízkém výsledku jedné z těchto metrik. Metrika F1 lze vypočítat pomocí vzorce: [18], [15]

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Tyto metriky jsem při experimentech počítal pro každou třídu výše uvedené metriky. Kompletní výsledky jsou uvedeny v přílohách A a B. Tato kapitola s výsledky neobsahuje výsledky pro třetí třídu (vedlejší provoz).

Experimenty spočívají ve změnách následujících parametrů:

- V neuronové síti jsem použil dva *typy buněk*: LSTM a GRU.
- Dále jsem měnil *počet prováděných epoch*: 1, 2, 5 a 10. Zdůvodu velkého množství dat (paketů) jsem neprováděl experimenty s větším počtem epoch. Při experimentech, kdy se nemění hodnota tohoto parametru je jeho základní hodnota 5.
- Měnil jsem *velikost okna*, což je počet paketů, které si neuronová síť pamatuje a využívá je pro klasifikaci aktuálně zpracovávaného paketu. Hodnoty: 1, 5, 10 a 25. Při experimentech, kde se nemění hodnota tohoto parametru je jeho základní hodnota 5.
- Jako poslední parametr jsem měnil *počet buněk* ve Forward a Backward části neuronové sítě (viz kapitola 3.3). Hodnoty: 2, 8, 16, 32, 64. Při experimentech, kdy se nemění hodnota tohoto parametru, je jeho základní hodnota 64.

Základní dělení experimentů je podle typu buněk v neuronové síti, tedy experimenty s buňkami LSTM a GRU.

Experimenty s LSTM buňkami při změnách parametrů: počet epoch a velikost okna

Nejprve jsem provedl experimenty s LSTM buňkami. Jako první experiment jsem provedl se změnami parametrů *počet epoch* a *velikost okna*. Zbylý parametr je nastaven na základní hodnotu. Kompletní výsledky tohoto experimentu jsou v příloze A v tabulkách A.1 až A.9.

Tabulky 5.1 a 5.2 obsahují výsledky metriky F1, které jsou shrnutím výsledků metrik precision a recall pro třídy SYN Flood útoku a legitimní pakety pouze s pakety SYN. Tabulka 5.1 obsahuje výsledky metriky F1, která využívá dat z tabulek A.1 a A.2. Nejlepších výsledků se dosahuje při vyšších hodnotách obou parametrů, tedy počet epoch i velikost okna. Největšího zvětšení výsledků se dosahuje při vyšším počtu prováděných epoch. Jak je vidět v tabulce i při velikosti okna 1 se s počtem epoch výsledky zvyšují. K podobným výsledkům dochází, když je počet epoch nastaven na hodnotu 1 a zvyšuje se velikost okna. Nejlepšího výsledku se dosahuje při nastavení hodnot neuronové sítě (neměnící se parametry mají základní hodnotu) s těmito hodnotami: počet epoch je rovno 10 a velikost okna je rovno 25.

Epochy / Velikost okna	1	2	5	10
1	58,4175%	84,6240%	92,7981%	89,5455%
5	96,0193%	97,4296%	99,0074%	99,0074%
10	97,3039%	98,3891%	98,6369%	99,0074%
25	97,9038%	98,0296%	98,5149%	99,2462%

Tabulka 5.1: Výsledky: F1 útoku, LSTM, změna počtu epoch a velikosti okna

Tabulka 5.2 obsahuje výsledky metriky F1, která využívá dat z tabulek A.4 a A.5. Výsledky jsou podobné jako v případě první tabulky, tedy výsledky jsou nejlepší při vyšších hodnotách obou parametrů.

Epochy / Velikost okna	1	2	5	10
1	17,3585%	85,9873%	96,1120%	98,7616%
5	77,3333%	78,3784%	96,5204%	99,6875%
10	70,8102%	88,1215%	99,0683%	99,6875%
25	71,9457%	90,2128%	99,5305%	98,7578%

Tabulka 5.2: Výsledky: F1 normálního provozu se SYN, LSTM, změna počtu epoch a velikosti okna

Při provedení aritmetického průměru buněk se stejnými parametry vychází nejlepší výsledek roven $99,34745\%$ s parametry: počet epoch je roven 10 a velikost okna je rovna hodnotám 5 nebo 10.

Experimenty s LSTM buňkami při změnách parametrů: počet epoch a počet buněk v síti

Tabulky 5.3 a 5.4 obsahují výsledky metriky F1, které jsou shrnutím výsledků pro útok a legitimní provoz pouze s příznaky SYN. Tabulka 5.3 obsahuje výsledky metriky F1, která využívá dat z tabulek A.10 a A.11. Nejlepších výsledků se opět dosahuje při vyšším počtu hodnot s obou parametrů. U těchto experimentů není oproti těm předchozím, tak velký rozdíl mezi výsledky s použitím pouze jedné epochy. Nejlepšího výsledku se dosahuje při nastavení parametrů, kde počet epoch se rovná hodnotě 10 a počet buněk se rovná hodnotě 32.

Epochy / Počet buněk	1	2	5	10
2	88,8641%	99,0074%	98,8848%	99,0074%
8	89,8649%	98,7624%	97,6744%	99,0074%
16	92,5581%	96,1538%	98,8848%	98,8848%
32	90,9920%	97,3171%	98,8848%	99,5012%
64	96,6102%	95,7983%	98,7624%	99,0074%

Tabulka 5.3: Výsledky: F1 útok, LSTM, změna počtu epoch a počet buněk ve vrstvách

Tabulka 5.4 obsahuje také výsledky metriky F1, která využívá dat z tabulek A.13 a A.14. V této tabulce lze vidět, že nejlepších výsledků dosahuje , když je počet epoch roven

10. Také je zde vidět, že při použití 10 epoch a různých počtech buněk se výsledky příliš neliší.

Epochy / Počet buněk	1	2	5	10
2	81,3864%	97,2561%	99,6875%	99,5320%
8	79,1563%	94,2393%	97,8462%	99,6875%
16	71,2054%	96,1948%	98,6090%	99,6875%
32	82,6144%	91,2230%	99,3769%	99,6875%
64	74,9706%	91,3295%	98,0031%	99,6875%

Tabulka 5.4: Výsledky: F1 normálního provozu se SYN, LSTM, změna počtu epoch a počet buněk ve vrstvách

Při provedení aritmetického průměru buněk se stejnými parametry vychází jako nejlepší výsledek roven 99, 59435% s parametry: počet epoch je roven 10 a počet buněk v síti je roven 32.

Experimenty s LSTM buňkami při změnách parametrů: počet buněk v síti a velikost okna

Tabulky 5.5 a 5.6 obsahují výsledky metriky F1, které jsou shrnutím výsledků pro útok a legitimní provoz pouze s příznaky SYN. Tabulka 5.5 obsahuje výsledky metriky F1, která využívá dat z tabulek A.19 a A.20. Zde se výsledky oproti všem předchozím experimentům mírně liší. Všechny výsledky v tabulce dosahují vysokých výsledků. Nejlepších výsledků je dosaženo při parametrech: hodnota velikosti okna je rovna 5 a počet buněk je roven hodnotě 16 a 32.

Velikost okna / Počet buněk	1	5	10	25
2	98,0198%	98,8651%	98,5185%	98,3891%
8	89,6000%	98,8848%	98,6403%	98,1459%
16	87,0056%	99,0074%	98,5185%	98,6403%
32	89,1648%	99,0074%	98,3931%	98,2630%
64	93,1707%	98,2456%	98,6403%	98,1459%

Tabulka 5.5: Výsledky: f1 útok, LSTM, změna velikosti okna a počet buněk ve vrstvách

Tabulka 5.6 obsahuje výsledky metriky F1, která využívá dat z tabulek A.22 a A.23. Stejně jako u předchozí tabulky jsou nejlepší výsledky v oblasti středních hodnot měněných parametrů. Tedy výsledky jsou nejlepší s parametry: počet buněk má hodnoty 8 nebo 16 a velikost okna má hodnoty 5 nebo 10.

Velikost okna / Počet buněk	1	5	10	25
2	97,4684%	98,4568%	99,5320%	99,0654%
8	98,4520%	99,6875%	99,6875%	99,3750%
16	97,2561%	99,6875%	99,5305%	99,5438%
32	98,4568%	99,2224%	99,5320%	99,0654%
64	95,7958%	99,5320%	99,5320%	99,2200%

Tabulka 5.6: Výsledky: f1 normálního provozu se SYN, LSTM, změna velikosti okna a počet buněk ve vrstvách

Při provedení aritmetického průměru buněk se stejnými parametry vychází jako nejlepší výsledek roven $99,34745\%$ s parametry: velikost okna je rovna 5 a počet buněk v síti je roven 16.

Experimenty s GRU buňkami při změnách parametrů: počet epoch a velikost okna

Stejné experimenty jsem provedl i pro typ buněk GRU. Tyto experimenty dosahují velice podobných výsledků jako experimenty s buňkami typu LSTM. Tedy nejlepší výsledků se dosahuje při vyšším počtu epoch a velikosti okna. Tabulky 5.7 a 5.8 obsahují výsledky metriky F1, které jsou shrnutím výsledků pro útok a legitimní provoz pouze s příznaky SYN. Tabulka 5.7 obsahuje výsledky metriky F1, která využívá dat z tabulek B.1 a B.2. Stejně jako u stejného experimentu, ale s buňkami LSTM dochází k největšímu zvětšení výsleků při větším počtu epoch. Stejně tak tomu je i při zvětšování velikosti okna.

Epochy / Velikost okna	1	2	5	10
1	19,9537%	96,0510%	98,9950%	98,3523%
5	83,0385%	98,2759%	99,0074%	99,3773%
10	98,0296%	98,2759%	98,3970%	99,5012%
25	96,6102%	97,9141%	98,8820%	99,1283%

Tabulka 5.7: Výsledky: f1 útoku, GRU, změna počtu epoch a velikosti okna

Tabulka 5.8 obsahuje výsledky metriky F1, která využívá dat z tabulek B.4 a B.5. Zde výsledky také odpovídají vzorci, kterým se řídí všechny předchozí experimenty. Tedy nejlepších výsledků je dosaženo při vyšších hodnotách měněných parametrů. Je zde ovšem malá výjimka a to, že velmi vysokého výsledku je dosaženo už při parametrech: počet epoch je roven 2 a velikost okna je rovna 10.

Epochy / Velikost okna	1	2	5	10
1	85,2405%	94,8328%	98,3051%	97,7029%
5	90,9613%	99,2224%	99,3769%	99,6875%
10	98,1538%	99,6875%	99,2151%	99,6875%
25	96,8037%	98,9114%	99,5305%	99,5305%

Tabulka 5.8: Výsledky: f1 normálního provozu se SYN, GRU, změna počtu epoch a velikosti okna

Při provedení aritmetického průměru buněk se stejnými parametry vychází jako nejlepší výsledek roven 99, 59435% s parametry: počet epoch je roven 10 a velikost okna je rovna 10.

Experimenty s GRU buňkami při změnách parametrů: počet epoch a počet buněk v síti

Tabulky 5.9 a 5.10 obsahují výsledky metriky F1, které jsou shrnutím výsledků pro útok a legitimní provoz pouze s příznaky SYN. Tabulka 5.9 obsahuje výsledky metriky F1, která využívá dat z tabulek B.10 a B.11. Zde je opět vidět, že nejlepších výsledků se dosahuje při vyšším počtu epoch, ale při nižším počtu buněk v síti. K této skutečnosti došlo u stejných experimentů, ale s buňkami typu LSTM.

Epochy / Počet buněk	1	2	5	10
2	97,5610%	98,6301%	99,1239%	99,0025%
8	98,0344%	98,7593%	99,0074%	99,7500%
16	80,0401%	99,0074%	98,8875%	99,7500%
32	94,7743%	98,5185%	98,7624%	99,5012%
64	97,3171%	98,7624%	99,0074%	98,7624%

Tabulka 5.9: Výsledky: f1 útok, GRU, změna počtu epoch a počet buněk ve vrstvách

Tabulka 5.10 obsahuje výsledky metriky F1, která využívá dat z tabulek B.13 a B.14. Výsledky v této tabulce ukazují, že v případě paketů patřících do legitimního provozu pouze s příznakem SYN příliš nezáleží na počtu buněk v síti.

Epochy / Počet buněk	1	2	5	10
2	90,5444%	99,2224%	99,2224%	98,9147%
8	88,6111%	99,0683%	99,6875%	99,6875%
16	83,1373%	99,6875%	99,3750%	99,6875%
32	96,2293%	98,4568%	99,5320%	99,5320%
64	98,7578%	99,3769%	99,6875%	99,6875%

Tabulka 5.10: Výsledky: f1 normálního provozu se SYN, GRU, změna počtu epoch a počet buněk ve vrstvách

Při provedení aritmetického průměru buněk se stejnými parametry vychází jako nejlepší výsledek roven 99, 71875% s parametry: počet epoch je roven 10 a počet buněk je roven 8 a 16.

Experimenty s GRU buňkami při změnách parametrů: počet buněk v síti a velikost okna

Tabulky 5.11 a 5.12 obsahují výsledky metriky F1, které jsou shrnutím výsledků pro útok a legitimní provoz pouze s příznaky SYN. Tabulka 5.11 obsahuje výsledky metriky F1, která využívá dat z tabulek B.19 a B.20. Stejně jako u buněk LSTM je nejlepší výsledek dosažen, když je počet buněk roven 8 a velikost okna rovna 10. Velmi vysokých výsledků se také ovšem dosahuje i u zbylých počtech buněk.

Velikost okna / Počet buněk	1	5	10	25
2	98,8736%	99,0074%	99,5012%	99,1304%
8	97,2081%	99,0074%	99,7500%	99,0074%
16	97,1357%	99,3773%	99,6255%	98,6403%
32	98,5075%	99,6255%	98,5149%	98,7624%
64	96,0289%	99,6255%	99,0074%	98,5149%

Tabulka 5.11: Výsledky: f1 útok, GRU, změna velikosti okna a počet buněk ve vrstvách

Tabulka 5.12 obsahuje výsledky metriky F1, která využívá dat z tabulek B.22 a B.23. Výsledky jsou velmi podobné výsledkům z předchozí tabulky. Největšího výsledku bylo dosaženo při nastavení parametrů: velikost okna je rovna 10 nebo 25 a počet buněk v síti je roven 8 nebo 16.

Velikost okna / Počet buněk	1	5	10	25
2	98,5959%	99,5320%	99,2175%	99,6865%
8	95,4545%	98,4568%	99,6875%	99,9923%
16	96,2382%	99,5320%	99,9923%	99,6865%
32	97,6526%	99,5305%	99,3769%	99,5305%
64	97,4441%	99,0683%	99,6875%	99,5305%

Tabulka 5.12: Výsledky: f1 normálního provozu se SYN, GRU, změna velikosti okna a počet buněk ve vrstvách

Při provedení aritmetického průměru buněk se stejnými parametry vychází jako nejlepší výsledek roven $99,8089\%$ s parametry: velikost okna je rovna 10 a počet buněk je roven 16.

Všechny provedené experimenty ukázaly, že na výsledek metrik precision i recall a tedy i F1 mají největší vliv parametry: počet provedených epoch a velikost okna. Typ použitých buněk v síti nebo počet těchto buněk výrazným způsobem neovlivňuje. Tato skutečnost lze zejména v tabulkách, při změnách velikosti okna a počtu buněk. V tabulkách jsou zvýrazněny nejvyšší hodnoty výsledků, ale i ostatní výsledky dosahují velmi vysokých hodnot. Při vyšším počtu provedených epoch a velikosti okna byl výsledek téměř vždy vyšší než 97%.

Kapitola 6

Závěr

Cílem bakalářské práce bylo navrhnout a implementovat konvenční algoritmický přístup a heuristický přístup s využitím neuronových sítí pro ochranu před DoS útoky typu SYN Flood. Práce má ukázat, jestli oba přístupy lze využít pro mitigaci těchto útoků.

V rámci řešení bakalářské práce jsem se nejprve seznámil s architekturou dnešních počítačových sítích. Nejvíce jsem se zaměřil na fungování protokolu TCP. Následně jsem se věnoval problematice kybernetických útoků typu odepření služby především jsem se zaměřil na útok zneužívající tento protokol SYN Flood. Poté jsem si nastudoval konvenční algoritmické přístupy určené pro mitigaci tohoto útoku. Pro svou práci jsem si zvolil ten, který je nejvhodnější pro zařízení pro ochranu před DoS útoky, které je vyvíjeno v rámci sdružení CESNET. V neposlední řadě jsem si nastudoval princip neuronových sítí a dohledal jsem využití těchto neuronových sítí v oblasti detekce SYN Flood útoků.

Po nastudování a získání vědomostí jsem provedl implementační návrh obou výše uvedených přístupů. Následně jsem nejprve na základě návrhu provedl implementaci konvenčního algoritimického přístupu. Implementace proběhla ve dvou fázích. V první fázi jsem si ověřil funkčnost tohoto přístupu a v druhé fázi jsem tuto implementaci rozšířil do formy modulu. Testování tohoto přístupu jsem provedl ve virtuálním prostředí a ověřil jsem si, že tato metoda funguje podle očekávání.

Poté jsem podle návrhu provedl implementaci neuronové sítě. Součástí této implementace bylo i vytvoření programu na úpravu vstupních dat pro neuronovou síť. Následně jsem provedl testování tohoto přístupu. Výsledky z testování ukazují, že lze tento přístup využít pro detekci SYN Flood útoků.

V současné době je implementovaný konvenční algoritmický přístup ve formě modulu integrován do zařízení pro ochranu před DoS útoky. Toto zařízení je experimentálně nasazeno v síťové infrastruktuře akademické sítě CESNET. Oba přístupy by mohly být použity v projektu Adaptivní ochrany před DDoS útoky. Dalším pokračováním této práce by mohlo být otestování použité neuronové sítě z pohledu efektivnosti, paměťové a výkonnostní složitosti. Také by mohla být provedena implementace heuristického přístupu v technologii programovatelných hradlových polí FPGA.

Literatura

[1] Keras: The Python Deep Learning library.
URL https://keras.io/

[2] Neuronové sítě. online, [Online; navštíveno 28.02.2019].
URL https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471

[3] scikit-learn: Machine Learning in Python.
URL https://scikit-learn.org/stable/

[4] Unb iscx intrusion detection evaluation dataset.
URL http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html

[5] CHARALAMPOS, P.: *Distributed Denial of Service Attacks*. National Technical University of Athens, [Online; navštíveno 20.01.2018].
URL https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-30/dos-attacks.html

[6] DURAČINSKÁ, Z.; BAŠTA, P.: DDoS - sofistikovaný útok nebo služba na objednávku? *IT Systems.*, ročník 4, 2015: s. 38–39, ISSN 1802-002X.

[7] EDDY, W.: *TCP SYN Flooding Attacks and Common Mitigations*. RFC 4987, [Online; navštíveno 20.01.2018].
URL https://tools.ietf.org/html/rfc4987

[8] KUČERA, J.: DDoS Protector User Manual. online.

[9] LIBEROUTER: Technologies. online.
URL https://www.liberouter.org/technologies/

[10] makaros: True Positive Rate and False Positive Rate (TPR, FPR) for Multi-Class Data in python [online]. Stack Overflow, 2018 [cit. 2019-02-05].
URL https://stackoverflow.com/questions/50666091/true-positive-rate-and-false-positive-rate-tpr-fpr-for-multi-class-data-in-py?answertab=active#tab-top

[11] MATOUŠEK, P.: *Síťové aplikace a jejich architektura*. Brno: VUTIUM, první vydání, 2014, ISBN 978-80-214-3766-1.

[12] ROKŮSEK, Z.: *Teoretické základy neuronových sítí*. JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH, Pedagogická fakulta, 2007.

[13] SIKORA, M.: *Detekce Slow-Rate DDoS útoků*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017.

[14] T., B.; A., P.: Enabling Reproducible Cyber Research – Four Labeled Datasets. 2016.

[15] ŠTURC, P.: Systém pro zpracování dat a vyhodnocení morfologické segmentace češtiny [online]. 2013 [cit. 2019-05-02].
URL https://is.muni.cz/th/p6zl1/

[16] VAN VEEN, F.: The Neural Network Zoo. online, [Online; navštíveno 28.02.2019].
URL http://www.asimovinstitute.org/neural-network-zoo/

[17] WESLEY., M. E.: Defenses Against TCP SYN Flooding Attacks. *The Internet Protocol Journal*, ročník 9, 2006.
URL https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-34/syn-flooding-attacks.html

[18] Wikipedia contributors: Precision and recall — Wikipedia, The Free Encyclopedia. 2019, [Online; accessed 11-May-2019].
URL https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=893227571

[19] Xiaoyong, Y.; Chuanhuang, L.; Xiaolin, L.: DeepDefense: Identifying DDoS Attack via Deep Learning. 2017: s. 1–8.

Příloha A

Tabulky výsledků LSTM

Epochy / Velikost okna	1	2	5	10
1	44,0915%	76,9388%	88,1166%	82,2547%
5	92,9907%	95,6731%	98,5185%	98,5185%
10	95,6627%	97,7833%	98,0296%	98,5185%
25	96,8293%	96,8370%	97,7887%	100,0000%

Tabulka A.1: Výsledky: precision útoku, LSTM, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	86,5337%	94,0150%	98,0050%	98,2544%
5	99,2519%	99,2519%	99,5012%	99,5012%
10	99,0025%	99,0025%	99,2519%	99,5012%
25	99,0025%	99,2519%	99,2519%	98,5037%

Tabulka A.2: Výsledky: recall útoku, LSTM, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	58,4175%	84,6240%	92,7981%	89,5455%
5	96,0193%	97,4296%	99,0074%	99,0074%
10	97,3039%	98,3891%	98,6369%	99,0074%
25	97,9038%	98,0296%	98,5149%	99,2462%

Tabulka A.3: Výsledky: f1 útoku, LSTM, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	9,5649%	87,3786%	95,3704%	97,5535%
5	63,0435%	64,4444%	99,0074%	99,3769%
10	54,8110%	78,7654%	98,1538%	99,3769%
25	56,1837%	82,1705%	99,0654%	97,5460%

Tabulka A.4: Výsledky: precision normálního provozu se SYN, LSTM, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	93,7304%	84,6395%	96,8652%	100,0000%
5	100,0000%	100,0000%	100,0000%	100,0000%
10	100,0000%	100,0000%	100,0000%	100,0000%
25	100,0000%	100,0000%	100,0000%	100,0000%

Tabulka A.5: Výsledky: recall normálního provozu se SYN, LSTM, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	17,3585%	85,9873%	96,1120%	98,7616%
5	77,3333%	78,3784%	96,5204%	99,6875%
10	70,8102%	88,1215%	99,0683%	99,6875%
25	71,9457%	90,2128%	99,5305%	98,7578%

Tabulka A.6: Výsledky: f1 normálního provozu se SYN, LSTM, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	100,0000%	100,0000%	100,0000%	100,0000%
5	100,0000%	100,0000%	100,0000%	100,0000%
10	100,0000%	100,0000%	100,0000%	100,0000%
25	100,0000%	100,0000%	100,0000%	100,0000%

Tabulka A.7: Výsledky: precision ostatní provoz, LSTM, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	87,7644%	99,6973%	99,8084%	99,6704%
5	99,1800%	99,2681%	99,8965%	99,9770%
10	98,9372%	99,6509%	99,9578%	99,9770%
25	99,0140%	99,6969%	99,9655%	99,9923%

Tabulka A.8: Výsledky: recall ostatní provoz, LSTM, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	93,4835%	99,8484%	99,9041%	99,8350%
5	99,5883%	99,6327%	99,9482%	99,9885%
10	99,4658%	99,8251%	99,9789%	99,9885%
25	99,5046%	99,8482%	99,9827%	99,9962%

Tabulka A.9: Výsledky: f1 ostatní provoz, LSTM, změna počtu epoch a velikosti okna

Epochy / Počet buněk	1	2	5	10
2	80,2817%	98,5185%	98,2759%	98,5185%
8	81,9302%	98,0344%	95,9135%	98,5185%
16	86,7102%	92,8074%	98,2759%	98,2759%
32	83,8235%	95,2267%	98,2759%	99,5012%
64	93,8824%	92,3611%	98,0344%	98,5185%

Tabulka A.10: Výsledky: precision útok, LSTM, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	99,5012%	99,5012%	99,5012%	99,5012%
8	99,5012%	99,5012%	99,5012%	99,5012%
16	99,2519%	99,7506%	99,5012%	99,5012%
32	99,5012%	99,5012%	99,5012%	99,5012%
64	99,5012%	99,5012%	99,5012%	99,5012%

Tabulka A.11: Výsledky: recall útok, LSTM, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	88,8641%	99,0074%	98,8848%	99,0074%
8	89,8649%	98,7624%	97,6744%	99,0074%
16	92,5581%	96,1538%	98,8848%	98,8848%
32	90,9920%	97,3171%	98,8848%	99,5012%
64	96,6102%	95,7983%	98,7624%	99,0074%

Tabulka A.12: Výsledky: f1 útok, LSTM, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	68,9130%	94,6588%	99,3769%	99,0683%
8	65,5031%	89,1061%	96,0725%	99,3769%
16	55,2860%	93,4911%	97,2561%	99,3769%
32	70,8520%	84,3085%	98,7616%	99,3769%
64	59,9624%	84,7185%	96,0843%	99,3769%

Tabulka A.13: Výsledky: precision normálního provozu se SYN, LSTM, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	99,3730%	100,0000%	100,0000%	100,0000%
8	100,0000%	100,0000%	99,6865%	100,0000%
16	100,0000%	99,0596%	100,0000%	100,0000%
32	99,0596%	99,3730%	100,0000%	100,0000%
64	100,0000%	99,0596%	96,0843%	100,0000%

Tabulka A.14: Výsledky: recall normálního provozu se SYN, LSTM, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	81,3864%	97,2561%	99,6875%	99,5320%
8	79,1563%	94,2393%	97,8462%	99,6875%
16	71,2054%	96,1948%	98,6090%	99,6875%
32	82,6144%	91,2230%	99,3769%	99,6875%
64	74,9706%	91,3295%	98,0031%	99,6875%

Tabulka A.15: Výsledky: f1 normálního provozu se SYN, LSTM, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	100,0000%	100,0000%	100,0000%	100,0000%
8	100,0000%	100,0000%	100,0000%	100,0000%
16	100,0000%	100,0000%	100,0000%	100,0000%
32	100,0000%	100,0000%	100,0000%	100,0000%
64	100,0000%	100,0000%	100,0000%	100,0000%

Tabulka A.16: Výsledky: precision ostatní provoz, LSTM, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	99,0918%	99,9157%	99,9732%	99,9732%
8	99,0267%	99,8276%	99,8965%	99,9770%
16	98,7891%	99,8122%	99,9464%	99,9732%
32	99,2259%	99,7126%	99,9828%	99,9923%
64	99,0918%	99,6743%	99,9272%	99,9770%

Tabulka A.17: Výsledky: recall ostatní provoz, LSTM, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	99,5438%	99,9578%	99,9866%	99,9866%
8	99,5110%	99,9137%	99,9482%	99,9885%
16	99,3909%	99,9060%	99,9732%	99,9866%
32	99,6115%	99,8561%	99,9828%	99,9962%
64	99,5438%	99,8369%	99,9636%	99,9885%

Tabulka A.18: Výsledky: f1 ostatní provoz, LSTM, změna počtu epoch a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	97,2973%	100,0000%	97,5550%	97,7833%
8	82,7004%	98,2759%	97,7941%	97,3039%
16	79,5455%	98,5185%	97,5550%	97,7833%
32	81,4433%	98,5185%	97,5490%	97,7778%
64	91,1695%	98,7406%	97,7941%	97,3039%

Tabulka A.19: Výsledky: precision útok, LSTM, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	98,7531%	97,7556%	99,5012%	99,0025%
8	97,7556%	99,5012%	99,5012%	99,0025%
16	96,0100%	99,5012%	99,5012%	99,0025%
32	98,5037%	99,5012%	99,2519%	98,7531%
64	95,2618%	97,7556%	99,5012%	99,0025%

Tabulka A.20: Výsledky: recall útok, LSTM, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	98,0198%	98,8651%	98,5185%	98,3891%
8	89,6000%	98,8848%	98,6403%	98,1459%
16	87,0056%	99,0074%	98,5185%	98,6403%
32	89,1648%	99,0074%	98,3931%	98,2630%
64	93,1707%	98,2456%	98,6403%	98,1459%

Tabulka A.21: Výsledky: f1 útok, LSTM, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	98,4026%	96,9605%	99,0683%	98,1481%
8	97,2477%	99,3769%	99,3769%	98,7578%
16	94,6588%	99,3769%	99,3750%	98,7578%
32	96,9605%	98,4568%	99,0683%	98,1481%
64	91,9308%	99,0683%	99,0683%	98,4520%

Tabulka A.22: Výsledky: precision normálního provozu se SYN, LSTM, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	96,5517%	100,0000%	100,0000%	100,0000%
8	99,6865%	100,0000%	100,0000%	100,0000%
16	100,0000%	100,0000%	99,6865%	100,0000%
32	100,0000%	100,0000%	100,0000%	100,0000%
64	100,0000%	100,0000%	100,0000%	100,0000%

Tabulka A.23: Výsledky: recall normálního provozu se SYN, LSTM, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	97,4684%	98,4568%	99,5320%	99,0654%
8	98,4520%	99,6875%	99,6875%	99,3750%
16	97,2561%	99,6875%	99,5305%	99,5438%
32	98,4568%	99,2224%	99,5320%	99,0654%
64	95,7958%	99,5320%	99,5320%	99,2200%

Tabulka A.24: Výsledky: f1 normálního provozu se SYN, LSTM, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	100,0000%	100,0000%	100,0000%	100,0000%
8	100,0000%	100,0000%	100,0000%	100,0000%
16	100,0000%	100,0000%	100,0000%	100,0000%
32	100,0000%	100,0000%	100,0000%	100,0000%
64	100,0000%	99,9770%	100,0000%	100,0000%

Tabulka A.25: Výsledky: precision ostatní provoz, LSTM, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	100,0000%	99,9962%	99,9578%	99,9578%
8	99,6896%	99,9732%	99,9655%	99,9578%
16	99,6130%	99,9770%	99,9655%	99,9655%
32	99,6398%	99,9655%	99,9616%	99,9616%
64	99,8237%	99,9808%	99,9616%	99,9540%

Tabulka A.26: Výsledky: recall ostatní provoz, LSTM, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	100,0000%	99,9981%	99,9789%	99,9789%
8	99,8446%	99,9866%	99,9827%	99,9789%
16	99,8061%	99,9885%	99,9827%	99,9866%
32	99,8196%	99,9828%	99,9808%	99,9808%
64	99,9118%	99,9789%	99,9808%	99,9770%

Tabulka A.27: Výsledky: f1 ostatní provoz, LSTM, změna velikosti okna a počet buněk ve vrstvách

Příloha B

Tabulky výsledků GRU

Epochy / Velikost okna	1	2	5	10
1	11,1239%	98,1771%	99,7468%	100,0000%
5	71,2500%	97,0803%	98,5185%	99,2537%
10	96,8370%	97,0803%	97,3171%	99,5012%
25	93,8824%	96,3768%	98,5149%	99,0050%

Tabulka B.1: Výsledky: precision útoku, GRU, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	96,7581%	94,0150%	98,2544%	96,7581%
5	99,5012%	99,5012%	99,5012%	99,5012%
10	99,2519%	99,5012%	99,5012%	99,5012%
25	99,5012%	99,5012%	99,2519%	99,2519%

Tabulka B.2: Výsledky: recall útoku, GRU, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	19,9537%	96,0510%	98,9950%	98,3523%
5	83,0385%	98,2759%	99,0074%	99,3773%
10	98,0296%	98,2759%	98,3970%	99,5012%
25	96,6102%	97,9141%	98,8820%	99,1283%

Tabulka B.3: Výsledky: f1 útoku, GRU, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	90,4930%	92,0354%	96,6667%	95,5090%
5	83,8624%	98,4568%	98,7616%	99,3769%
10	96,3746%	99,3769%	99,3711%	99,3769%
25	93,8053%	97,8462%	99,0654%	99,0654%

Tabulka B.4: Výsledky: precision normálního provozu se SYN, GRU, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	80,5643%	97,8056%	100,0000%	100,0000%
5	99,3730%	100,0000%	100,0000%	100,0000%
10	100,0000%	100,0000%	99,0596%	100,0000%
25	100,0000%	100,0000%	100,0000%	100,0000%

Tabulka B.5: Výsledky: recall normálního provozu se SYN, GRU, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	85,2405%	94,8328%	98,3051%	97,7029%
5	90,9613%	99,2224%	99,3769%	99,6875%
10	98,1538%	99,6875%	99,2151%	99,6875%
25	96,8037%	98,9114%	99,5305%	99,5305%

Tabulka B.6: Výsledky: f1 normálního provozu se SYN, GRU, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	100,0000%	100,0000%	100,0000%	100,0000%
5	100,0000%	100,0000%	100,0000%	100,0000%
10	100,0000%	100,0000%	100,0000%	100,0000%
25	100,0000%	100,0000%	100,0000%	100,0000%

Tabulka B.7: Výsledky: precision ostatní provoz, GRU, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	88,3047%	99,9885%	99,9808%	99,9923%
5	99,1646%	99,9425%	99,9693%	99,9885%
10	99,9156%	99,9540%	99,9693%	100,0000%
25	99,8274%	99,9233%	99,9770%	99,9847%

Tabulka B.8: Výsledky: recall ostatní provoz, GRU, změna počtu epoch a velikosti okna

Epochy / Velikost okna	1	2	5	10
1	93,7892%	99,9943%	99,9904%	99,9962%
5	99,5806%	99,9713%	99,9847%	99,9943%
10	99,9578%	99,9770%	99,9847%	99,9962%
25	99,9136%	99,9616%	99,9885%	99,9923%

Tabulka B.9: Výsledky: f1 ostatní provoz, GRU, změna počtu epoch a velikosti okna

Epochy / Počet buněk	1	2	5	10
2	95,4654%	98,5075%	99,4975%	99,0025%
8	96,6102%	98,2716%	98,5185%	100,0000%
16	66,9463%	98,5185%	98,0392%	100,0000%
32	90,4762%	97,5550%	98,0344%	99,5012%
64	95,2267%	98,0344%	98,5185%	98,0344%

Tabulka B.10: Výsledky: precision útok, GRU, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	99,7506%	98,7531%	98,7531%	99,0025%
8	99,5012%	99,2519%	99,5012%	99,5012%
16	99,5012%	99,5012%	99,7506%	99,5012%
32	99,5012%	99,5012%	99,5012%	99,5012%
64	99,5012%	99,5012%	99,5012%	99,5012%

Tabulka B.11: Výsledky: recall útok, GRU, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	97,5610%	98,6301%	99,1239%	99,0025%
8	98,0344%	98,7593%	99,0074%	99,7500%
16	80,0401%	99,0074%	98,8875%	99,7500%
32	94,7743%	98,5185%	98,7624%	99,5012%
64	97,3171%	98,7624%	99,0074%	98,7624%

Tabulka B.12: Výsledky: f1 útok, GRU, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	83,3773%	98,4568%	98,4568%	97,8528%
8	79,5511%	98,1538%	99,3769%	99,3769%
16	71,3004%	99,3769%	99,0654%	99,3769%
32	92,7326%	96,9605%	99,0683%	99,0683%
64	97,8462%	98,7616%	99,3769%	99,3769%

Tabulka B.13: Výsledky: precision normálního provozu se SYN, GRU, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	99,0596%	100,0000%	100,0000%	100,0000%
8	100,0000%	100,0000%	100,0000%	100,0000%
16	99,6865%	100,0000%	99,6865%	100,0000%
32	100,0000%	100,0000%	100,0000%	100,0000%
64	99,6865%	100,0000%	100,0000%	100,0000%

Tabulka B.14: Výsledky: recall normálního provozu se SYN, GRU, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	90,5444%	99,2224%	99,2224%	98,9147%
8	88,6111%	99,0683%	99,6875%	99,6875%
16	83,1373%	99,6875%	99,3750%	99,6875%
32	96,2293%	98,4568%	99,5320%	99,5320%
64	98,7578%	99,3769%	99,6875%	99,6875%

Tabulka B.15: Výsledky: f1 normálního provozu se SYN, GRU, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	100,0000%	100,0000%	100,0000%	100,0000%
8	100,0000%	100,0000%	100,0000%	100,0000%
16	100,0000%	100,0000%	100,0000%	100,0000%
32	100,0000%	100,0000%	100,0000%	100,0000%
64	100,0000%	100,0000%	100,0000%	100,0000%

Tabulka B.16: Výsledky: precision ostatní provoz, GRU, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	99,7011%	99,9770%	99,9923%	99,9732%
8	99,6398%	99,9617%	99,9770%	100,0000%
16	98,7661%	99,9770%	99,9655%	100,0000%
32	99,7509%	99,9310%	99,9655%	99,9885%
64	99,9080%	99,9617%	99,9770%	99,9693%

Tabulka B.17: Výsledky: recall ostatní provoz, GRU, změna počtu epoch a počet buněk ve vrstvách

Epochy / Počet buněk	1	2	5	10
2	99,8503%	99,9885%	99,9962%	99,9866%
8	99,8196%	99,9808%	99,9885%	100,0000%
16	99,3792%	99,9885%	99,9828%	100,0000%
32	99,8753%	99,9655%	99,9828%	99,9943%
64	99,9540%	99,9808%	99,9885%	99,9847%

Tabulka B.18: Výsledky: f1 ostatní provoz, GRU, změna počtu epoch a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	99,2462%	98,5185%	99,5012%	98,7624%
8	98,9664%	98,5185%	100,0000%	98,5185%
16	97,0149%	99,2537%	98,5149%	97,7941%
32	98,2630%	99,7500%	97,7887%	98,0344%
64	92,7907%	99,7500%	98,5185%	97,7887%

Tabulka B.19: Výsledky: precision útok, GRU, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	98,5037%	99,5012%	99,5012%	99,5012%
8	95,5112%	99,5012%	99,5012%	99,5012%
16	97,2569%	99,5012%	99,2519%	99,5012%
32	98,7531%	99,5012%	99,2519%	99,5012%
64	99,5012%	99,5012%	99,5012%	99,2519%

Tabulka B.20: Výsledky: recall útok, GRU, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	98,8736%	99,0074%	99,5012%	99,1304%
8	97,2081%	99,0074%	99,7500%	99,0074%
16	97,1357%	99,3773%	99,6255%	98,6403%
32	98,5075%	99,6255%	98,5149%	98,7624%
64	96,0289%	99,6255%	99,0074%	98,5149%

Tabulka B.21: Výsledky: f1 útok, GRU, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	98,1366%	99,0683%	99,0625%	100,0000%
8	92,3754%	96,9605%	99,3769%	99,3750%
16	96,2382%	99,0683%	99,0683%	99,3750%
32	97,5000%	99,3750%	98,7616%	99,0654%
64	99,3485%	98,1538%	99,3769%	99,0654%

Tabulka B.22: Výsledky: precision normálního provozu se SYN, GRU, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	99,0596%	100,0000%	99,3730%	100,0000%
8	98,7461%	100,0000%	100,0000%	100,0000%
16	96,2382%	100,0000%	100,0000%	100,0000%
32	97,8056%	99,6865%	100,0000%	100,0000%
64	95,6113%	100,0000%	100,0000%	100,0000%

Tabulka B.23: Výsledky: recall normálního provozu se SYN, GRU, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	98,5959%	99,5320%	99,2175%	99,6865%
8	95,4545%	98,4568%	99,6875%	99,9923%
16	96,2382%	99,5320%	99,9923%	99,6865%
32	97,6526%	99,5305%	99,3769%	99,5305%
64	97,4441%	99,0683%	99,6875%	99,5305%

Tabulka B.24: Výsledky: f1 normálního provozu se SYN, GRU, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	100,0000%	100,0000%	100,0000%	100,0000%
8	100,0000%	100,0000%	100,0000%	100,0000%
16	100,0000%	100,0000%	100,0000%	100,0000%
32	100,0000%	100,0000%	100,0000%	100,0000%
64	100,0000%	100,0000%	100,0000%	100,0000%

Tabulka B.25: Výsledky: precision ostatní provoz, GRU, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	100,0000%	99,9732%	99,9962%	99,9808%
8	99,9693%	99,9464%	100,0000%	99,9770%
16	99,9962%	99,9847%	99,9770%	99,9655%
32	99,9885%	100,0000%	99,9616%	99,9655%
64	99,9349%	99,9808%	99,9770%	99,9655%

Tabulka B.26: Výsledky: recall ostatní provoz, GRU, změna velikosti okna a počet buněk ve vrstvách

Velikost okna / Počet buněk	1	5	10	25
2	100,0000%	99,9866%	99,9981%	99,9904%
8	99,9847%	99,9732%	100,0000%	99,9885%
16	99,9981%	99,9923%	99,9923%	99,9827%
32	99,9943%	100,0000%	99,9808%	99,9827%
64	99,9674%	99,9904%	99,9885%	99,9827%

Tabulka B.27: Výsledky: f1 ostatní provoz, GRU, změna velikosti okna a počet buněk ve vrstvách