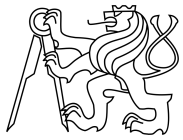
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Analýza síťového provozu s pomocí komunikačních map
Student:	Bc. Tomáš Vicher
Vedoucí:	Ing. Tomáš Čejka
Studijní program:	Informatika
Studijní obor:	Počítačové systémy a sítě
Katedra:	Katedra počítačových systémů
Platnost zadání:	Do konce zimního semestru 2016/17

Pokyny pro vypracování

Nastudujte způsob monitorování síťového provozu na základě sledování síťových toků.
Navrhněte metodu zpracování informací o síťových tocích k vygenerování a udržování grafu komunikace mezi jednotlivými uzly sítě popř. agregovaných uzlů (podsítí).
Navrhněte využití vzniklých grafových struktur ke sledování vývoje provozu na síti, případně k detekci neobvyklých jevů na síti.
Navrženou detekční metodu implementujte jako modul do systému Nemea.
Otestujte tento mechanismus za pomoci simulace příp. s použitím reálného provozu z počítačové sítě.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Tomáš Zahradnický, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 12. června 2015

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Diplomová práce

Analýza síťového provozu s pomocí komunikačních map

Bc. Tomáš Vicher

Vedoucí práce: Ing. Tomáš Čejka

9. května 2016

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Tomáši Čejkovi za rady a čas, které mi věnoval. Dále bych rád poděkoval rodině a přítelkyni za podoporu při práci i po celou dobu mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 9. května 2016 .....

Odkaz na tuto práci

Vicher, Tomáš. *Analýza síťového provozu s pomocí komunikačních map.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato diplomová práce popisuje návrh a implementaci metody pro zpracování informací o síťových tocích. Informace slouží ke generování grafu komunikace mezi jednotlivými uzly sítě, případně agregovanými uzly (podsítě). Stav grafu je udržován aktuální díky vyhodnocování informací v téměř reálném čase. Součástí práce byl návrh využití takto vzniklých struktur s ohledem na sledování vývoje provozu, případně k detekci neobvyklých jevů v síti. Výsledkem práce je detekční modul implementovaný pro systém NEMEA. Součástí práce je též analýza různých možností monitorování sítě, síťových útoků a statistických metod pro vyhodnocování informací ze síťových toků.

Klíčová slova síťová analýza, síťový tok, Holt-Wintersova metoda, Python, monitorování, graf

Abstract

The master thesis describes design and implementation of method for processing information about network flow. The information is used to generate and preserve graph for communication between network nodes or agregated nodes (subnetworks). The graph is kept in current state by almost real time information processing. The thesis includes design of use for graph structures

for traffic monitoring and anomaly detection. The result of the master thesis is the detection method implemented as a module to the NEMEA system. The analysis of network monitoring, network attacks and statistical methods for processing network informations are included.

Keywords network analysis, network flow, Holt-Winters method, Python, monitoring, graph

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Spojení a výměna informací mezi počítači a dalšími zařízeními v rámci počítačové sítě s sebou přináší četné hrozby v podobě různých způsobů útoků, zneužití nebo selhání zařízení.

Na síťový provoz v lokálních sítích mohou být směřovány útoky přicházející jak z internetu, tak z vnitřní sítě (DOS, DDOS, skenování portů, šíření červů mezi napadenými zařízeními). Tyto útoky mohou mít různý dopad na funkcionalitu sítě, například:

- nedostupnost služeb
- proniknutí škodlivého softwaru
- rozesílání nežádoucího obsahu nebo zpráv do dalších zařízení
- odcizení nebo ztráta citlivých a jinak důležitých dat

Stejné důsledky může způsobit ať už úmyslná či neúmyslná chyba osob pracujících se zařízeními a síťovou infrastrukturou, nebo selhání některého zařízení či služeb.

Jednou z metod pro vyhodnocování problémů popsaných výše je analýza chování sítě s pomocí síťových toků. K analýze chování a případné detekci změny v chování lze využít vyhodnocování počtu toků [6]. Podle [7] se zvýšením počtu toků projevuje skenování portů, DOS, DDOS, nebo šíření červů. Dalšími příklady jsou anomálie ve zvýšení počtu toků u napadených zařízení, které vytváří zvýšené množství síťové komunikace. Sníženým počtem toků se může projevit selhání některé služby nebo problém v síťovém spojení.

Motivací k vypracování této práce bylo pomoci IT administrátorům s vyhodnocováním informací ze sítě a detekcí případných anomálií, proto byl

v rámci této práce vytvořen modul pro systém NEMEA[1] určený pro nasazení v lokálních sítích.

Modul vyhodnocuje anomálie v počtu toků na základě předpovědi budoucích hodnot s ohledem na denní a týdenní sezónnost. Jeho další funkcí je vyhodnocení změny ve struktuře sítě v podobě odpojování a připojování zařízení, vytváření a zanikání spojení mezi zařízeními s rozlišením na známá a neznámá zařízení. To může s hlášením anomálie v počtu toků ukázat například připojení útočníka do sítě.

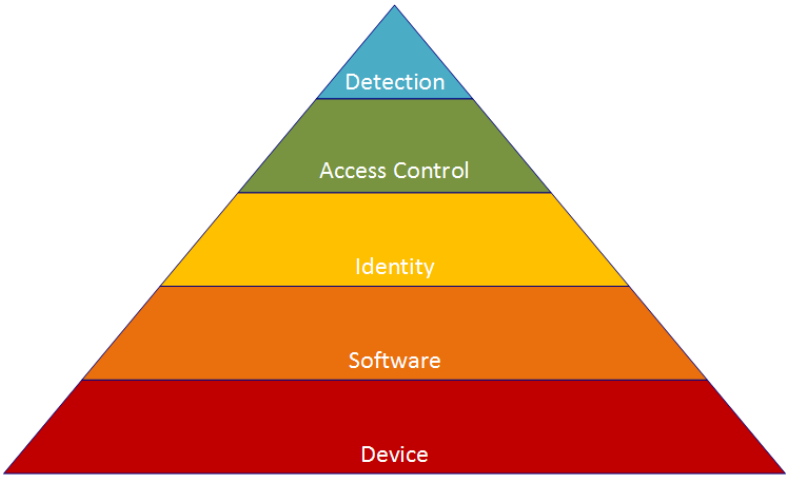
Anomálie jsou vyhodnocovány na úrovni jednotlivých zařízení, spojení mezi zařízeními i pro celkový provoz ve sledovaném síťovém rozsahu. Vyhodnocené informace slouží k nahlášení problému a upřesnění jeho lokalizace pro následné prověření zařízení, na kterém je anomálie detekována.

Pro kontrolu struktury sítě modul exportuje graf zařízení a spojení mezi zařízeními. Tento graf lze využít například k vizuální kontrole, nebo k prezentaci sítě osobám, které strukturu neznají.

Testování modulu proběhlo na datech z testovací sítě u vedoucího práce. Nasazení modulu s vyhodnocováním dat z provozu v reálném čase bylo testováno ve firemním prostředí.

V obrázku 1 je zařazeno monitorování a vyhodnocování dat ze sítě do celkového přehledu zabezpečení v IT. Článek [1] popisuje jednotlivé vrstvy zabezpečení. První vrstva zobrazuje správu zařízení a inventarizaci, následuje správa a údržba softwaru. Další vrstvou je kontrola nad identitou osob. Na předchozí vrstvy zabezpečení navazuje řízení přístupu ke zdrojům a informacím. Na vrcholu této bezpečnostní pyramidy se nachází monitorování a detekce - téma na které je zaměřena tato diplomová práce.

[1] NEMEA je systém fungující na základě modulů umožňující monitorování a vyhodnocování dat ze sítě. Další informace a zdrojové kódy jsou na adrese `https://github.com/CESNET/NEMEA`



Obrázek 1: Hierarchie zabezpečení v IT. Zdroj: [1]

Struktura práce

Práce je strukturována do tematicky oddělených kapitol.
V první části je uvedena specifikace cílů této práce.
Druhá část popisuje analýzu zpracování síťových toků, vyhodnocení síťového provozu, anomálií v síti a statistických metod. Výstupem jsou informace pro třetí část práce.
Třetí část popisuje návrh řešení pro vyhodnocování síťového provozu s využitím síťových toků. Popsány jsou použité nástroje, struktura a funkce navrhovaného modulu.
Ve čtvrté části jsou popsány možnosti napojení modulů do systému NEMEA.
Pátá část se věnuje popisu implementace navrženého řešení jako modulu do systému NEMEA.
V šesté části je popsáno testování implementovaného modulu s daty z testovací sítě u vedoucího práce a nasazení a testování ve firemní síti.
V závěrečné části je uvedena diskuze, zhodnocení splněných cílů práce a možnosti budoucího rozšíření.

Kapitola 1

Specifikace cílů

Cílem práce je navrhnout a implementovat nástroj, který umožní administrátorům v lokálních sítích sledovat stav sítě a detekovat případné anomálie. Nástroj má být navržen jako modul pro systém NEMEA. Nástroj umožňuje vyhodnocovat data ze síťových toků a detekovat anomálie na úrovni jednotlivých zařízení a spojů mezi zařízeními. Po nahlášení anomálií lze problém lokalizovat a na postiženém zařízení provést podrobnější analýzu, například kontrolou systémových logů, další analýzu provozu, kontrolu firewallu. Nástroj je navrhován pro zpracování a ukládání informací ze síťových toků. Z těchto informací jsou vytvářeny komunikační mapy sítě, zaznamenané jako orientovaný graf. Zařízení jsou při vyhodnocování identifikována podle IP adres získaných ze zpracovávaných záznamů o tocích. Data jsou vyhodnocována dvěma způsoby. Jedním je vyhodnocování provozu na základě množství síťových toků, druhým je vyhodnocování připojování a odpojování zařízení. Z vyhodnocování množství síťových toků lze téměř v reálném čase odhalit chování sítě, které může ukazovat na probíhající útok, selhání funkce některého zařízení, spojení, nebo jinou změnu v provozu, která se projevuje zvýšeným nebo sníženým množstvím toků oproti běžnému provozu. Hlášené detekce lze sledovat textově uložené v logovacím souboru nebo v grafické podobě. Z grafu sítě vytvářeného při zpracování dat nástroj umožní exportovat graf připojených zařízení a spojení mezi zařízeními. Zobrazení grafu sítě poskytne přehled nad připojenými zařízeními a spojení mezi zařízeními, ze kterého lze sledovat, jestli jsou připojena ta zařízení, která mají být a komunikace probíhá mezi zařízeními, kde má probíhat. Tyto grafy mohou posloužit k prezentaci sítě osobám, které strukturu sítě neznají.

Vyhodnocování anomálií v počtu síťových toků se doplňuje s vyhodnocováním připojení známých a neznámých zařízení do sítě, odpojení ze sítě, vznikem nebo zaniknutím spojení mezi zařízeními. Příkladem může být nahlášení připojení neznámého zařízení a následné hlášení anomálie ve zvýšení počtu toků, které způsobilo nově připojené zařízení. Známá a neznámá zařízení a spojení jsou určena pomocí seznamu známých zařízení a seznamu známých

spojení mezi zařízeními, které je za běhu aplikace možné upravovat. Hlášena jsou připojení a odpojení zařízení ze sítě a vznikající a zanikající spojení mezi zařízeními. Nástroj reprezentuje výsledky vyhodnocování ve více úrovních podrobnosti, podle nastavení uživatelem.

Kapitola 2

Analýza

Tato kapitola se zabývá možnostmi monitorování sítě a vyhodnocování informací ze síťového provozu. Zaměřuje se především na vyhodnocování informací ze síťových toků. V úvodu kapitoly je popsáno srovnání existujících řešení s navrhovaným. V souvislosti s monitorováním a vyhodnocováním toků je zahrnuta sekce věnující se systému NEMEA, pro který byl navrhnut a implementován modul. Dále je obsažena sekce zabývající se problémem detekce anomálií v síti, obsahující některé způsoby detekce anomálií s využitím zpracování síťových toků. Uvedeny jsou některé statistické metody pro zpracování časových řad a použití k detekci anomálií v síťovém provozu.

2.1 Srovnání s existujícím řešením

Modulární systém NEMEA pro analýzu síťového provozu je stále vyvíjen. V aktuální chvíli jsou v systému dostupné následující moduly pro detekci anomálií v síti zaměřené na konkrétní druhy útoků:

- amplification_detection: Modul pro detekci amplifikačních útoků.[8] [9]
- blacklistfilter: Modul kontrolující pozorované IP adresy na přítomnost ve veřejných černých listinách.
- hoststatsnemea: Modul sbírající statistiky ze síťového provozu, ze kterých vyhodnocuje zařízení skenující síť, DoS útočníky a oběti, SSH brute-force útoky a DNS amplifikační útoky.[10]
- sip_bf_detector: Modul pro detekci brute-force útoků pokoušejících se prolomit hesla uživatelů SIP zařízení.[11]
- tunnel_detection: Modul pro detekci DNS tunelů.[12]
- voip_fraud_detection: Modul pro detekci testování prefixů ve VoIP, učení prefixů zemí, do kterých zařízení volá a detekci volání do jiné země. [13]

- vportscan_detector: Modul pro detekci vertikálního skenování portů.[14]

Modul vytvořený v této práci je navržen pro využití v lokálních sítích. Modul vyhodnocuje a předpovídá počty toků pro IP adresy, spojení mezi IP adresami a celý vyhodnocovaný síťový rozsah. Oproti modulům které detekují a hlásí konkrétní druh útoku, přináší navrhovaný modul detekci a hlášení anomálií v počtu toků na jednotlivých IP adresách a spojeních mezi adresami, orientovanými podle zdrojové a cílové IP adresy. K detekci modul využívá předpovědi následujících hodnot Holt-Wintersovou metodou se dvěma sezónami 2.6.7. Detekované anomálie mohou být způsobeny různými druhy útoků jako šíření červů, DOS útok, skenování portů [15], nebo jiných problémů se síťovou infrastrukturou a zařízeními, které se projevují zvýšením nebo snížením počtu toků. Na základě hlášení anomálie může být provedeno prověření zařízení, nebo spojení, na kterém byla anomálie hlášena. Výhodou navrženého modulu je možnost spuštění ve výchozím nastavení, protože jsou parametry potřebné pro předpověď budoucích hodnot i výpočet prahových hodnot pro detekci anomálie počítány automaticky za chodu. Modul dále provádí detekci připojení a odpojení známých a neznámých IP adres a spojení mezi adresami s možností aktualizace seznamů známých adres a spojení, kterou výše zmíněné moduly nenabízejí. Tyto seznamy jsou načítány a aktualizovány v interních strukturách za běhu modulu.

2.2 Komunikační mapy

Komunikační mapy jsou v této práci zaznamenávány jako orientovaný graf vytvářený z informací ze záznamů o síťových tocích. Orientovaný graf `G` je definován jako trojice složená z uzlů `V(G)`, hran `E(G)` a funkce přiřazující každé hraně uspořádanou dvojici uzlů.[16]

Uzly jsou v této práci tvořeny IP adresami, orientované hrany jsou dvojice zdrojové a cílové IP adresy získané ze záznamu o toku. K uzlům a hranám jsou asociovány další datové struktury využívané v modulu.

2.3 Vyhodnocování síťového provozu

Monitorování a vyhodnocování síťového provozu lze podle způsobu zpracování síťových dat rozdělit na analýzu paketů a analýzu síťových toků. Na základě způsobu detekce lze vyhodnocování rozdělit na detekci útoků pomocí signatur a detekci anomálií na základě chování sítě.[17] S těmito způsoby monitorování a vyhodnocování síťového provozu pracují bezpečnostní systémy IDS - systémy detekce průniku a IPS - systémy prevence průniku.

- IDS je bezpečnostní systém monitorující počítačové systémy a síťový provoz, provádějící analýzu dat za účelem odhalení útoků a zneužití sítě z vnějšího i vnitřního prostředí organizace.[18]

- IPS je bezpečnostní systém s funkcemi definovanými v IDS, poskytující aktivní zahazování paketů nebo ukončení síťového spojení, které obsahuje neautorizovaná data.[19]

IDS/IPS se dělí na základě vyhodnocovaných dat a umístění v síti na HIDS/HIPS a NIDS/NIPS [17]:

- HIDS/HIPS jsou systémy vyhodnocující data z jednotlivých zařízení jako využití zdrojů, logy systému a další informace získané z operačního systému.
- NIDS/NIPS jsou systémy monitorující a vyhodnocující provoz v síti.

Tato práce je dále orientována na NIDS/NIPS. V tabulce 1 jsou zpracovány hlavní odlišnosti NIPS a NIDS podle [20]:

Tabulka 1: Srovnání NIDS a NIPS.

NIPS	NIDS
Pracuje jako mezičlánek v síťovém provozu.	Analyzuje kopii síťového provozu.
Aby nedošlo ke zpomalení,musí zvládat zpracovat síťový provoz v reálném čase.	Nezpomaluje síťový provoz.
Zabraňuje průniku škodlivého provozu do sítě.	Nebrání přímým zásahem průniku škodlivého provozu do sítě.

Hlavní rozdíl ve funkčnosti mezi NIPS a NIDS je podle [20], že NIPS zabraňuje proniknutí škodlivého provozu, zatímco NIDS zpracovává kopii provozu a při detekci sám neprovádí opatření k zablokování útoku.

2.3.1 Signatury

Informace získané z vyhodnocování síťového provozu mohou být využity k detekci útoků na základě signatur. Signatury jsou podle [20] definovány jako soubory pravidel, které jsou využívány k detekci útoků v síťovém provozu. Detekční systém porovnává síťový provoz se známými útoky a hlásí nebo blokuje útok v případě shody v porovnání. Útok který není v databázi signatur není detekován. Z tohoto důvodu je potřeba pravidelná aktualizace signatur. Článek [20] popisuje rozdělení bezpečnostních systémů IDS a IPS založených na signaturách společnosti Cisco na základě druhů signatur. Systémy jsou rozděleny do hlavních komponent podle kategorií signatur, které jsou dále rozděleny na mikrokomponenty vyhodnocující signatury rozdělené do podrobnějších kategorií:

- Komponenta atomických signatur: Obsahuje mikrokomponenty pro TCP, UDP, ICMP a další.

- Komponenta signatur služeb: Obsahuje mikrokomponenty pro HTTP, DNS, FTP a další.
- Komponenta signatur řetězců: Obsahuje mikrokomponenty pro signatury řeřezců s regulárními výrazy. Příklad signatury pro detekci dat obsahujících soubory s koncovkou `.exe`, `.com`, `.bat` může vypadat takto: `".*\.([Ee][Xx][Ee]`[Cc][Oo][Mm]|[Bb][Aa][Tt])".
- Komponenta signatur multi řetězců: Obsahuje signatury kontrolující více řetězců.
- Komponenta ostatních signatur: Zpracovává ostatní signatury nezařazené do předchozích kategorií.

Níže je uvedeno několik útoků a možností jak využít signatury k jejich detekci podle [21]:

- Detekce připojení IP adresy, která je rezervovaná pro jiné zařízení. Lze odhalit kontrolou zdrojové adresy v hlavičce paketu oproti signatuře IP adres.
- Detekce paketů s neočekávaným nastavením příznaků. Lze odhalit porovnáním nastavení příznaků paketů se signaturou správných a špatných nastavení příznaků paketu.
- Detekce nežádoucího doručeného emailu. Lze odhalit porovnáním obsahu předmětu, nebo jiných částí zprávy se signaturou obsahující nežádoucí záznamy.
- Detekce pokusu o DNS přetečení zásobníku. Lze odhalit parsováním DNS zprávy na jednotlivá políčka a kontrolou jejich délky oproti signatuře obsahující očekávaný formát.
- Detekce DOS útoku který je realizován pomocí velkého množství stejných dotazů nebo dotazů obsahujících nějaký vzor. Lze odhalit porovnáním se signaturou obsahující hledané vzory v paketech a počítáním množství příchozích paketů identifikovaných podle signatury.

2.3.2 Chování sítě

Analýza chování sítě (behaviorální analýza) je technika detekce průniku využívající vzorů chování vyhodnocených ze síťového provozu. Tato technika slouží pro identifikaci možných útoků nebo technických problémů s minimálním rizikem ohrožení soukromí dat v síti. Analýza není založena na obsahu ale na statistikách z provozu.[22]

Článek [20] uvádí některé výhody a nevýhody ze srovnání systémů využívajících signatur a systémů vyhodnocujících chování sítě.

Výhodou signaturového systému oproti systému analýzy chování sítě je méně falešně pozitivních detekcí - detekce útoku, který ve skutečnosti neprobíhá. [23] Nevýhodou je, že systém nedetekuje události, na které nemá uloženy signatury a vyžaduje doplňování nových signatur.

Výhodou analýzy chování sítě oproti signaturám je detekce neznámých útoků. Nevýhodou je nutnost předem určit chování v síti, které je považováno za normální a slouží jako reference při vyhodnocování.

2.3.3 Analýza paketů

Jednou z možností monitorování síťového provozu je analýza jednotlivých paketů. Při paketové analýze jsou zachytávány a analyzovány všechny pakety procházející bodem pozorování. Analýza může probíhat na datech z hlavičky a obsahu paketu. Z dat zapouzdřených v paketu lze analyzovat informace z druhé až sedmé vrstvy OSI modelu[24]. Pro detekci útoků s analýzou paketů je obvykle využívána metoda detekce pomocí signatur.[17]

2.3.4 Analýza toků

Síťový tok je podle IPFIX od IETF [25] definován jako sada IP paketů procházejících pozorovaným místem v síti v určeném časovém intervalu. Všechny pakety patřící do jednoho toku se vyznačují množinou shodných parametrů. Každý z těchto parametrů toku je vyhodnocen z následujících parametrů paketu:

- jedno nebo více políček z hlavičky paketu (např. cílová IP adresa), políčka z transportní části hlavičky paketu(např. číslo cílového portu), nebo políčka z aplikační části hlavičky (např. RTP políčka [26])
- jedna nebo více charakteristik samotného paketu (např. počet MPLS návěští[27])
- jedno nebo více políček odvozených ze zpracovaných paketů (např. IP adresa dalšího hopu)

V IPFIX terminologii jsou pak používané parametry nazývány klíči. Formát toku složený z pěti klíčů může vypadat například takto:

```
(ip_src,ip_dst,port_src,port_dst,protocol)
```

Nový tok je vytvořen v případě, že je zaznamenán paket, který nemá shodné všechny parametry s některým z aktivních toků a nelze jím některý aktivní tok aktualizovat. Záznam toku je ukončen a exportován, pokud nastane některá z následujících situací:

- Je vyhodnocen konec toku detekováním FIN nebo RST bitu v TCP spojení.

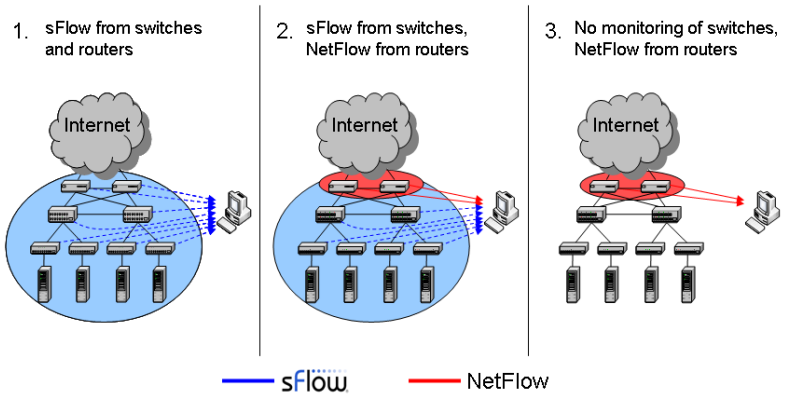
- Tok je neaktivní po stanovený čas (nebyl aktualizován záznamem nového paketu).
- Tok překročí stanovený maximální časový interval.
- Nastane interní chyba v exportéru.

Pro vyhodnocování informací ze síťových toků je podle [17] obvykle využívána detekce anomálií na základě chování sítě. Sledovat a vyhodnocovat anomálie lze ze síťových spojů mezi počítači, síťovými počítačovými periferiemi, například síťovými tiskárnami a jinými síťovými zařízeními. Vyhodnocovat lze různé statistiky provozu (množství přenesených dat, počet přenesených toků, počet přenesených paketů a další statistiky).

Zpracování toků nabízí oproti paketové analýze nižší náročnost na paměť, které je dosaženo tím, že zpracování paketů vyžaduje operace s větším množstvím dat, než zpracování toků. Do toku může být sdruženo více paketů a obsažena jen část informace z paketů, které sdružuje. Do toku se nezahrnuje například datová a další části, které nejsou uvedeny ve specifikaci formátu toku.[17] Díky odlišným vlastnostem analýzy toků a paketů lze využít oba mechanizmy tak, aby se vzájemně doplnily.

2.3.5 Technologie monitorování sítí

V praxi jsou pro monitorování sítí využity různé technologie. V článku [2] jsou některé (NetFlow, sFlow, JFlow, NetStream, IPFIX) uvedeny. Zásadní odlišnosti se nachází mezi sFlow a ostatními jmenovanými. SFlow je založené na vyhodnocování paketů, NetFlow a ostatní jmenované na vyhodnocování toků. SFlow operuje na druhé L2 (linkové) vrstvě ISO/OSI síťového modelu. NetFlow a další jmenované technologie fungují na třetí L3 a čtvrté L4 vrstvě ISO/OSI. V obrázku 2je vidět typické nasazení a vzájemné doplnění sFlow a NetFlow



Obrázek 2: Nasazení NetFlow a sFlow. Zdroj: [2]

Ve třech scénářích je shrnuto typické využití a vzájemné doplnění sFlow a NetFlow:

1. Všechny přepínače a směrovače podporují sFlow. Data přepínačů a směrovačů sbírá a vyhodnocuje centrální zařízení, které poskytuje přehled nad všemi zapojenými prvky.

2. Přepínače podporují sFlow a směrovače NetFlow. Stává se tak typicky v situacích, kdy jsou přepínače od jiného výrobce než směrovače. Síťová infrastruktura zůstává celá monitorována.

3. Přepínače nepodporují monitorování provozu, směrovače podporují NetFlow. Monitorování zařízení a provozu je omezeno pouze na směřovače.

Dále jsou uvedeny informace o NetFlow/IPFIX architektuře, která je podporována kolektorem IPFIXcol vyvíjeným sdružením CESNET. NetFlow je protokol vytvořený firmou Cisco Networks, který definuje služby pro sběr dat ze sítě, zpracování do síťových toků a administraci sítě s pomocí síťových toků[28]. IPFIX je v [29] popsán jako protokol vycházející z NetFlow, jehož funkcionalitu rozšiřuje. IPFIX přináší oproti NetFlow dvě zásadní rozšíření:

- IPFIX umožňuje zaznamenávat vendor ID. Tím může výrobce síťového hardwaru přidat své informace do toku. Tento postup umožňuje přenášet v tocích informace, které obvykle zpracovávají monitorovací systémy pomocí SNMP, nebo jsou ukládány do syslogu.

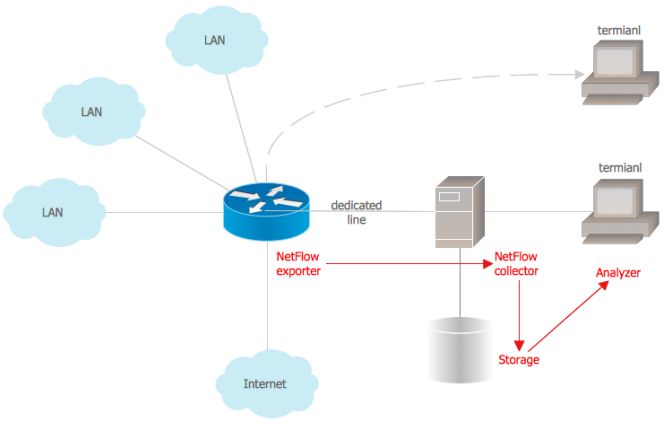
- IPFIX oproti NetFlow povoluje proměnnou délku políček

V obrázku 3 je zakresleno základní schéma NetFlow architektury obsahující exportér, kolektor a analyzér.

Exportér je podle IETF [28] popsán jako zařízení (například směřovač) podporující NetFlow služby. Exportér monitoruje pakety vstupující do pozorovaného bodu a z těchto paketů vytváří toky. Informace exportuje v podobě záznamu toku do kolektoru.

NetFlow kolektor je zařízení, které přijímá záznamy toků z jednoho nebo více exportérů. Obdržené záznamy zpracovává a ukládá informace o záznamu toku. Ukládané záznamy mohou být před uložením ještě agregovány. Data z kolektoru zpracovává třetí prvek v nákresu architektury, analyzér.

Analyzér je aplikace, která vyhodnocuje výsledná data, případně vytváří reprezentaci dat pro uživatele.[3]



Obrázek 3: Architektura NetFlow. Zdroj: [3]

2.4 Anomálie v síťovém provozu

Detekce anomálií v síťovém provozu je problémem nalezení neobvyklých vzorů chování v síti, které neodpovídají očekávanému normálnímu chování.[30] Anomálie může být detekována jako změna v rozdělení pozorované náhodné veličiny. V sekvenci dat $y_{1,n} = (y_1, \ldots, y_n)$ nastává změna v čase $t \in \{1, \ldots, n-1\}$, když mezi $y_1, \ldots, y_t$ a $y_{t+1}, \ldots, y_n$ existuje změna v rozdělení[31].

2.4.1 Detekce na základě chyby předpovědi

Jedena z metod pro detekci anomálií v časových řadách je popsaná v [32], [33]. Navržena je třemi na sebe navazujícími částmi:

- Model pro předpověď hodnoty jednoho následujícího časového okna časové řady.
- Měření odchylky mezi předpovězenou hodnotou a pozorovanou hodnotou - chyby v předpovědi.
- Mechanizmus pro rozhodnutí, jestli a v jakém čase přesáhla chyba předpovědi prahovou hodnotu pro určení anomálie.

Záznam abnormálně velké chyby předpovědi ukazuje na změnu ve sledované náhodné veličině. Chyby předpovědi jsou nazývány reziduály, protože představují míru variability která není součástí modelu pro předpověď.[33]

Reziduál e je podle [34] definován pro každou hodnotu vyhodnocované sekvence dat jako:

$$e = y - \hat{y}$$

kde
y je pozorovaná hodnota,
$\hat{y}$ je předpovězená hodnota.

Článek [32] popisuje detekční metodu založenou na využitím Holt-Wintersovy metody pro predikci budoucí hodnoty. Holt-Wintersova a některé další metody založené na klouzavém průměru umožňující předpověď hodnoty v časové řadě jsou popsány v sekci 2.6.

Metoda využívá exponenciálního vyhlazování reziduálů:

$$d_t = \gamma|y_t - \hat{y}_t| + (1 - \gamma)d_{t-m}$$

kde:
d_t je exponenciálně vyhlazený reziduál v čase t,
γ je vyhlazovací konstanta sezónnosti v Holt-Wintersově metodě,
y_t je naměřená hodnota v čase t,
$\hat{y}_t$ je předpovězená hodnota v čase t,
d_{t-m} je exponenciálně vyhlazený reziduál v čase $t - m$.

Anomálie je vyhodnocována na základě pozorované hodnoty y_t a intervalu určeného jako:

$$(\hat{y}_t - \delta_- \cdot d_{t-m}, \hat{y}_t + \delta_+ \cdot d_{t-m})$$

kde:
y_t je naměřená hodnota v čase t,
$\hat{y}_t$ je předpovězená hodnota v čase t,
δ_- a δ_+ jsou konstanty pro úpravu šířky intervalu,
d_{t-m} je exponenciálně vyhlazený reziduál v čase $t - m$.

V případě, že pozorovaná hodnota y_t nenáleží výše zmíněnému intervalu, je událost považována za anomálii. Při vyhodnocení stavu jako anomálie nastává riziko dvou druhů chyb[23]:

- Falešně pozitivní výsledek. Nastává při vyhodnocení legitimního provozu jako anomálie.
- Falešně negativní výsledek. Nastává při nenahlášení anomálie v době výskytu útoku, na jehož detekci je bezpečnostní systém stavěn.

Nastavením hodnot δ_- a δ_+ lze měnit šířku intervalu. Pokud má mít horní a dolní část intervalu stejnou velikost, musí být zvoleno $\delta_- = \delta_+$

Článek uvádí dvě možnosti, kdy detekovat anomálie. Jednou možností je detekovat anomálii pokaždé, když naměřená hodnota leží vně intervalu. Druhou možností je určit pohyblivé okno pevné velikosti pro pozorované hodnoty a

prahovou hodnotu počtu měření spadající vně intervalu. V případě že je v pohyblivém okně tato hodnota překročena, je hlášena anomálie. Tato možnost slouží ke snížení počtu falešně pozitivních hlášení anomálií.

Takto lze vyhodnocovat data sbíraná ze síťového provozu, agregovaná do oken časové řady. Sledovat lze různé statistiky z provozu v síti (množství přenesených paketů, množství přenesených síťových toků, počet přenesených bajtů a další statistiky podle druhu vyhodnocovaných dat). Tento postup je využit v kapitole návrhu.

2.4.2 Sekvenční metody

Dalším způsobem vyhodnocení anomálií je použití některé ze sekvenčních metod pro detekci bodu změny. Jednou ze sekvenčních metod je Pageova CUSUM, definovaná v [35]. Metoda funguje na základě testování hypotézy H_0 že nastala změna v rozdělení v čase t a alternativní hypotézy H_1 že změna nenastala, $t = \infty$. Testování probíhá a základě výpočtu:

$$Z_{n,t} = \sum_{k=t}^{n} log \frac{p_1(y_k|y_1, y_2, \ldots, y_k - 1)}{p_0(y_k|y_1, y_2, \ldots, y_k - 1)}, n \geq t$$

Metoda je založena na porovnání statistiky $U_n = \max_{1 \leq t \leq n}(Z_n, t)$ s prahovou hodnotou h. Anomálie je hlášena pokud je překročena hodnota h:

$$\tau_{CU}(h) = \min\{n \geq 1 : U_n \geq h\}$$

Metoda CUSUM je další možností jak detekovat anomálie na základě vyhodnocení reziduálů definovaných výše v 2.4.1. Touto problematikou se zabývá článek[33].

2.5 Síťové útoky

V této sekci jsou popsány některé druhy síťových útoků uvedené v článku [36] a jejich typické projevy jako síťové anomálie. Vybrány jsou útoky, které se mohou projevovat zvýšením počtu toků v provozu.

2.5.1 Skenování

Skenování portů se může projevovat velkým množstvím malých paketů, které mohou vygenerovat zvýšené množství toků. Skenování portů lze rozdělit na tři kategorie:

- vertikální skenování
- horizontální skenování
- blokové skenování

Při vertikálním skenování dochází ke skenování více portů na jednom zařízení, které se projeví zvýšením počtu toků na skenovaném zařízení. Horizontální skenování je určeno skenováním jednoho portu napříč sítí. Nárůst počtu toků může být sledován na celkovém množství toků ve skenované síti, rozloženém mezi jednotlivá zařízení. Blokový sken je kombinací předchozích dvou.

2.5.2 Odmítnutí služby (DOS)

Mezi DOS útoky patří více druhů útoků. Pro detekování některých druhů DOS je vhodnější využití analýzy chování sítě, pro některé je vhodnější využití signatur. DOS útoky lze rozdělit na sémantické útoky a útoky přetížením požadavky. Mezi lépe rozpoznatelné typy DOS pomocí analýzy chování sítě patří útoky, kde je podstatou přetížení zdroje.

Detekce s využitím signatur je vhodnější u útoků založených na principu sémantického útoku. V těchto případech nedochází k odmítnutí služby na základě zahlcení, ale kvůli obsahu zprávy. Příkladem je útok nazývaný Ping of Death. Útočník odesílá záměrně poškozené ping pakety, které způsobí pád systému. Nemožnost identifikace tohoto útoku pomocí analýzy chování sítě je způsobena tím, že je při tomto útoku použit pouze jeden ICMP[2] tok a nedochází tím k výkyvům statistik chování sítě.

2.5.3 Červi

Napadení systému červem bývá složeno ze dvou částí. V první fázi proběhne prozkoumání sítě za účelem nalezení zranitelnosti a proniknutí do systému. V druhé části je červ přenesen do systému. V tomto procesu je pomocí sledování síťových toků možné odhalit první fázi průzkumu sítě, která se může projevovat podobným způsobem, jako skenování sítě. Detekce přenosu červa do systému může být odhalena kontrolou obsahu paketů

2.6 Metody zpracování časových řad

Tato sekce se zabývá metodami pro zpracování časových řad. Časová řada je podle [37] sekvence dat měřená v určitém časovém období, definovaná jako množina vektorů $x_{(t)}$, $t = 0, 1, 2, \ldots$, kde t představuje uplynulý čas. $x_{(t)}$ je náhodná veličina a jednotlivá měření jsou chronologicky uspořádána.

Popsány jsou metody umožňující zpracování časových řad a předpověď budoucích hodnot, které lze využít při detekci anomálií. Rozebrána je metoda jednoduchého klouzavého průměru a jednoduchého exponenciálního vyhlazování, metody dvojitého a trojitého exponenciálního vyhlazování a metody počítajících s trendem a sezónností[38].

[2]Protokol používaný pro oznamování chyb nebo nedostupnosti služby.

2.6.1 Jednoduchý klouzavý průměr

Metoda jednoduchého klouzavého průměru je založena na výpočtu aritmetického průměru z předchozích období zpracovávané časové řady. Vzorec pro výpočet jednoduchého klouzavého průměru je:

$$SMA = \frac{y_1 + y_2 + \cdots + y_n}{n} \tag{2.1}$$

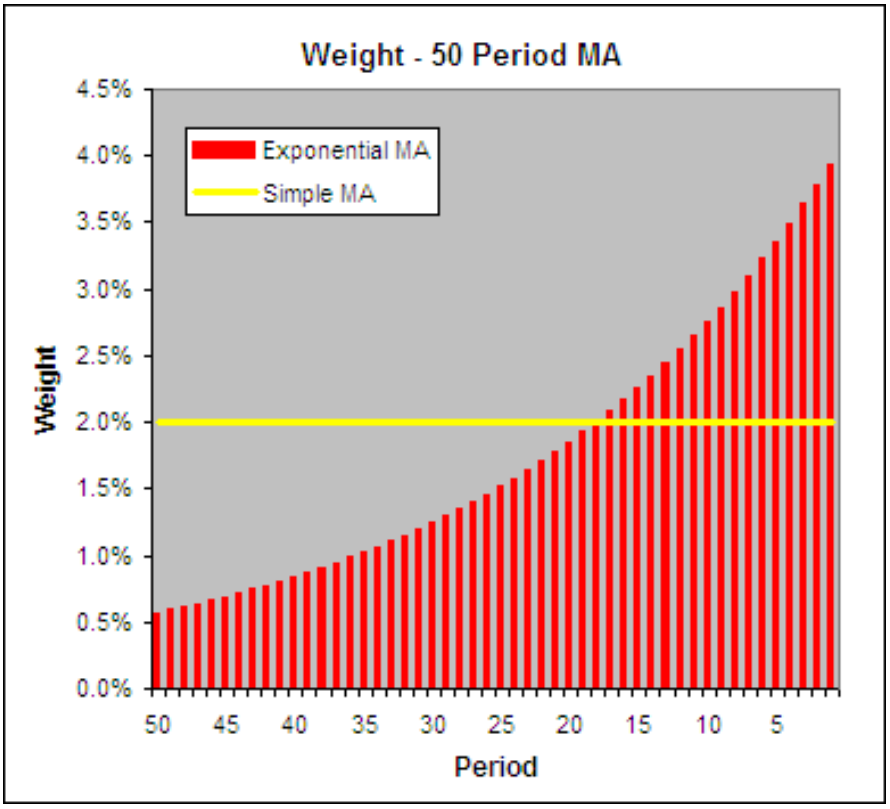
Předpovězenou hodnotu v čase t lze vyjádřit jako:

$$\hat{y}_{t+1} = \frac{y_t + y_{t-1} + \cdots + y_{t-n+1}}{n} \tag{2.2}$$

y_{t-n+1} až y_t jsou hodnoty z časové řady,
$\hat{y}_{t+1}$ je předpovězená hodnota pro čas $t + 1$.

2.6.2 Exponenciální vyhlazování

Exponenciální vyhlazování (exponenciální klouzavý průměr) je obdobně jako jednoduchý klouzavý průměr založeno na výpočtu z předchozích pozorovaných hodnot - z časové řady. Tato metoda oproti jednoduchému klouzavému průměru počítá s exponenciálně klesající váhou dat směrem do minulosti. Článek [39] uvádí oproti jednoduchému klouzavému průměru vyšší citlivost na poslední pozorované hodnoty. Tím je způsobena rychlejší reakce na změny hodnot. Na obrázku 4 je vidět ukázka srovnání váhy u exponenciálního vyhlazování s konstantní váhou u jednoduchého klouzavého průměru.



Obrázek 4: Váha při výpočtu jednoduchého klouzavého průměru a exponenciálního klouzavého průměru. Zdroj [4]

Z časové řady je u exponenciálního vyhlazování sestavován trend, který lze vyjádřit jako polynom k-tého stupně vzorcem:

$$y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots + \beta_k t^k \tag{2.3}$$

S využitím vzorce trendu lze získat vzorec pro výpočet konkrétní pozorované hodnoty v čase jako:

$$y_{t-j} = T_{t-j}^{(k)} + \epsilon_{t-j} = \beta_0 - \beta_1 j + \beta_2 j^2 - \cdots + (-1)^k \beta_k j^k + \epsilon_{t-j} \tag{2.4}$$

kde:
y_{t-j} je pozorovaná hodnota,
$t = 1, 2, \ldots, n$ je index aktuální pozice v pozorování,
$j = 0, 1, \ldots, t-1$ je index stáří pozorování vztažený k indexu v okamžiku pozorování,
ϵ je náhodná složka.

U exponenciálního vyhlazování jsou přiřazeny k jednotlivým hodnotám ve zpracovávané časové řadě váhy, které směrem do vzdálenější minulosti expo-

nenciálně klesají. Odhad parametrů lze vyjádřit váženou metodou nejmenších čtverců:

$$\min_{\beta_0,\ldots,\beta k} \sum_{j=0}^{t-1} a^j (y_{t-j} - T_{t-j}^{(k)})^2 \tag{2.5}$$

kde $0 < \alpha < 1$, tudíž α^j s rostoucím j klesá. Z tohoto vzorce jsou odvozeny parametry $\beta_0, \ldots, \beta_k$. Tímto postupem jsou parametry odvozovány znova z aktuálních dát při každém opakování pozorování. Pro snížení časové náročnosti mohou být využity vyhlazovací statistiky, sestavené z počátečních odhadů parametrů $\beta_0, \ldots, \beta_k$, které jsou dále rekurentně aktualizovány.

2.6.3 Jednoduché exponenciální vyhlazování

V jednoduchém exponenciálním vyhlazování je trend považován za konstantní:

$$y_{t-j} = T_{t-j} = \beta_0, j = 0, 1, \ldots, t-1 \tag{2.6}$$

Hodnoty vyhlazování v čase t jsou vypočteny ze vzorce:

$$\hat{y}_t = (1-\alpha) y_t + \alpha \hat{y}_{t-1}, t = 1, \ldots, n \tag{2.7}$$

Počáteční hodnota $\hat{y}_0$ je určena jako aritmetický průměr:

$$\hat{y}_0 = \frac{1}{n} \sum_{t=1}^{n} y_t \tag{2.8}$$

Bodová předpověď v čase t pro čas $t+T, T = 1, 2, ...$ označená $\hat{y}_{t+T}^P$ je vypočítána jako:

$$\hat{y}_{t+T}^P = \hat{y}_t \tag{2.9}$$

2.6.4 Dvojité exponenciální vyhlazování

V metodě dvojitého exponenciálního vyhlazování se uvažuje lineární trend:

$$y_{t-j} = T_{t-j} = \beta_0 - \beta_1 j, j = 0, 1, \ldots, t-1 \tag{2.10}$$

Pro výpočet vyhlazovacích statistik jsou použity tyto vzorce:

$$S_t = (1-\alpha) y_t + \alpha S_{t-1} \tag{2.11}$$

$$S_t^{[2]} = (1-\alpha) S_t + \alpha S_{t-1}^{[2]} \tag{2.12}$$

Počáteční hodnoty S_0 a $S_0^{[2]}$ jsou získány z odhadnutých parametrů $\beta_0(0)$ a $\beta_1(0)$:

$$S_0 = \hat{\beta}_0(0) - \frac{\alpha}{1-\alpha} \hat{\beta}_1(0) \tag{2.13}$$

$$S_0^{[2]} = \hat{\beta}_0(0) - \frac{2\alpha}{1-\alpha} \hat{\beta}_1(0) \tag{2.14}$$

kde $\beta_0(0)$ a $\beta_0^{[2]}(0)$ jsou spočítány metodou nejmenších čtverců.

Vyhlazená hodnota v čase t je spočítána vzorcem:

$$\hat{y}_t = 2S_t - S_t^{[2]} \tag{2.15}$$

Bodová předpověď v čase t pro čas $t+T, T = 1, 2, \ldots$ bude:

$$\hat{y}_{t+T}^P = (2 + \frac{(1-\alpha)T}{\alpha})S_t - (1 + \frac{(1-\alpha)T}{\alpha})S_t^{[2]} \tag{2.16}$$

2.6.5 Trojité exponenciální vyhlazování

Trojité exponenciální vyhlazování pracuje s kvadratickým trendem:

$$y_{t-j} = T_{t-j} = \beta_0 - \beta_1 j + \beta_2 j^2, j = 0, 1, \ldots, t-1 \tag{2.17}$$

Vyhlazovací statistiky lze zapsat rovnicemi:

$$S_t = (1-\alpha)y_t + \alpha S_{t-1} \tag{2.18}$$
$$S_t^2 = (1-\alpha)S_t + \alpha S_{t-1}^2 \tag{2.19}$$
$$S_t^3 = (1-\alpha)S_t^2 + \alpha S_{t-1}^3 \tag{2.20}$$

Počáteční hodnoty pro vyhlazovací statistiky jsou vypočítány rovnicemi:

$$S_0 = \hat{\beta}_0(0) - \frac{\alpha}{1-\alpha}\hat{\beta}_1(0) + \frac{\alpha(1+\alpha)}{2(1-\alpha)^2}\hat{\beta}_2(0) \tag{2.21}$$
$$S_0^{[2]} = \hat{\beta}_0(0) - \frac{2\alpha}{1-\alpha}\hat{\beta}_1(0) + \frac{2\alpha(1+2\alpha)}{2(1-\alpha)^2}\hat{\beta}_2(0) \tag{2.22}$$
$$S_0^{[3]} = \hat{\beta}_0(0) - \frac{3\alpha}{1-\alpha}\hat{\beta}_1(0) + \frac{3\alpha(1+3\alpha)}{2(1-\alpha)^2}\hat{\beta}_2(0) \tag{2.23}$$

β_0, β_1, β_2 jsou získány metodou nejmenších čtverců.
Vyhlazené hodnoty v čase t jsou vypočítány vzorcem:

$$\hat{y}_t = 3S_t - 3S_t^{[2]} + 3S_t^{[3]} \tag{2.24}$$

Bodová předpověď v čase t pro čas $t+T, T = 1, 2, \ldots$ bude:

$$\hat{y}_{t+T}^P = \frac{1}{2\alpha^2}\{[6\alpha^2 + (1+5\alpha)(1-\alpha)T + (1-\alpha)^2T^2]S_t - \tag{2.25}$$
$$- [6\alpha^2 + 2(1+4\alpha)(1-\alpha)T + 2(1-\alpha)^2T^2]S_t^{[2]} + \tag{2.26}$$
$$+ [2\alpha^2 + (1+3\alpha)(1-\alpha)T + (1-\alpha)^2T^2]S_t^{[3]}\} \tag{2.27}$$

2.6.6 Holtova metoda s lineárním trendem

Holtova metoda exponenciálního vyhlazování trendů (zpracována 1957) rozšiřuje jednoduché exponenciální vyhlazování. Pracuje se dvěma vyhlazovacími konstantami α a β. Konstanta α slouží k adaptivnímu odhadu úrovně β_0 v čase t a β pro adaptivní odhad směrnice lineárního trendu β_1 v čase t.

Odhad parametrů β_0 a β_1 v čase t je:

$$\hat{\beta}_{0,t} = \alpha y_t + (1-\alpha)(\hat{\beta}_{0,t-1} + \hat{\beta}_{1,t-1}) \tag{2.28}$$

$$\hat{\beta}_{1,t} = \beta(\hat{\beta}_{0,t} - \hat{\beta}_{0,t-1}) + (1-\beta)\hat{\beta}_{1,t-1} \tag{2.29}$$

kde:
$\hat{\beta}_{0,t}$ je odhad úrovně lineárního trendu v čase t,
$\hat{\beta}_{1,t}$ je odhad směrnice lineárního trendu v čase t,
$\hat{\beta}_{0,t-1}$ je odhad úrovně lineárního trendu v čase $t-1$,
$\hat{\beta}_{1,t-1}$ je odhad směrnice lineárního trendu v $t-1$,
$\alpha \in \langle 0, 1\rangle$ je vyhlazovací konstanta úrovně,
$\beta \in \langle 0, 1\rangle$ je vyhlazovací konstanta směrnice.
Bodová předpověď s horizontem období $h > 0$ v čase t je definována:

$$\hat{y}_t(h) = \hat{\beta}_{0,t} + h\hat{\beta}_{1,t} \tag{2.30}$$

Pro $h = 1$ v čase $t-1$ jsou bodová předpověď a chyba předpovědi:

$$\hat{y}_{t-1}(1) = \hat{\beta}_{0,t-1} + \hat{\beta}_{1,t-1} \tag{2.31}$$

$$\hat{a}_t = y_t - \hat{\beta}_{0,t-1} - \hat{\beta}_{1,t-1} \tag{2.32}$$

2.6.7 Holt Wintersova metoda se sezónními trendy

Winters rozšířil Holtovu metodu exponenciálního vyhlazování trendů o aditivní a multiplikativní sezónnost. Sezónnost je pro aditivní a multiplikativní formu vyjádřena rovnicemi:

$$y_t = (\beta_0 + \beta_1 t) + S_t + a_t \tag{2.33}$$

$$y_t = (\beta_0 + \beta_1 t) S_t a_t \tag{2.34}$$

kde:
β_0 je parametr úrovně lineárního trendu,
β_1 je parametr směrnice lineárního trendu,
t je časová proměnná,
S_t je sezónní průměr nebo sezónní index v čase t,
přičemž musí platit:

1. pro sezónní průměry $\sum_{j=1}^{s} S_j = 0$,

2. pro sezónní indexy $\sum_{j=1}^{s} S_j = s$,

a_t je nesystematická složka typu bílého šumu.
Rekurentně lze Holt-Wintersovu metodu aditivního typu zapsat následujícími rovnicemi:

$$\hat{y}_t(h) = l_t + hb_t + s_{t-m+h} \tag{2.35}$$
$$l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \tag{2.36}$$
$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \tag{2.37}$$
$$s_t = \gamma(y_t - l_t) + (1-\gamma)s_{t-m} \tag{2.38}$$

kde:
$\hat{y}_t(h)$ je bodová předpověď s horizontem $h > 0$ v čase t,
l_t je odhad úrovně lineárního trendu v čase t,
b_t je odhad směrnice lineárního trendu v čase t,
s_t je odhad sezónního výkyvu v čase t.
m určuje periodu sezóny

V článku [40] je popsáno rozšíření metody o druhou sezónnost. Bodová předpověď pro aditivní metodu bude:

$$\hat{y}_t(h) = l_t + hb_t + s_{t-m1+h} + s^{[2]}_{t-m^{[2]}+h} \tag{2.39}$$

Oproti variantě s jednou sezónou je změna v rovnici pro výpočet sezónnosti s_t a přibyla rovnice pro druhou sezónu $s^{[2]}_t$:

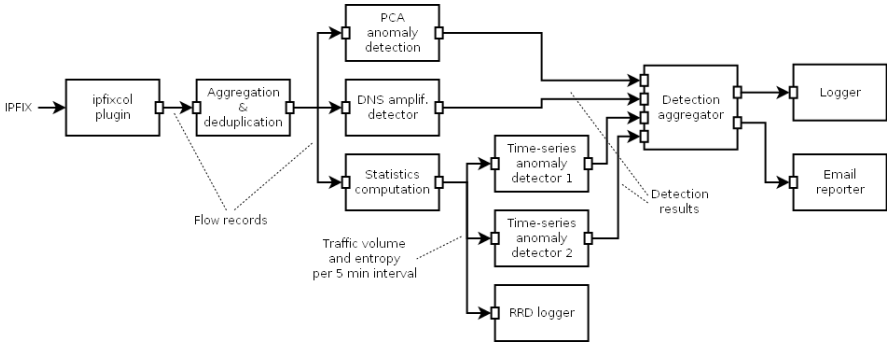
$$s_t = \gamma(y_t - l_t - s^{[2]}_{t-m^{[2]}}) + (1-\gamma)s_{t-m} \tag{2.40}$$
$$s^{[2]}_t = \omega(y_t - l_t - s_{t-m}) + (1-\omega)s^{[2]}_{t-m^{[2]}} \tag{2.41}$$

kde $m^{[2]}$ je perioda druhé sezónnosti,
ω je vyhlazovací konstanta pro druhou sezónu.

2.7 NEMEA

2.7.1 Struktura systému

NEMEA je framework, který umožňuje sestavení IDS systému z jednotlivých modulů. Moduly spolu komunikují přes síťová rozhraní. Data mezi moduly jsou přenášena pomocí TRAP platformy ve formátu, který specifikuje protokol UniRec. Zpracování dat v NEMEA probíhá v reálném čase. Data se posílají jako proudy z výstupních do vstupních rozhraní. Načítání dat je možné provádět z IPFIX kolektoru [3] nebo číst předem připravená uložená data. V obrázku 5 je zobrazen NEMEA systém v sestavení pro komplexní monitorování sítě.



Obrázek 5: Příklad NEMEA sestavení systému. Zdroj: [5]

Aktivní jsou moduly pro sběr dat, zpracování dat, vyhodnocení a detekci anomálií, zpracování a report výsledků. Formát zasílaných dat je mezi moduly dohodnut při připojení modulu do systému. Modul tak může přijímat a odebírat pouze potřebná data.

2.7.2 Moduly

Každý modul funguje v systému jako samostatný proces. Níže jsou vypsány některé výhody návrhu modulů jako samostatných procesů.

- Moduly mohou být spuštěny a zastaveny v různém čase nezávisle na ostatních.
- Moduly lze spravovat ze strany operačního systému nezávisle na dalších částech NEMEA systému.
- K implementaci modulů je možné použít různé programovací jazyky.
- Každý modul má jako samostatný proces přiděleny systémové zdroje.
- V případě pádu procesu se nevyřadí z provozu celý NEMEA systém.
- Moduly lze samostatně aktualizovat.

Moduly zachovávají jednotnost ve formátu dat a v použití rozhraní. Tím udržují kompatibilitu a možnost zapojení do NEMEA systému. Moduly mohou mít více vstupních i výstupních rozhraní a mohou být spojeny s více moduly. V případě ztráty spojení se pokouší o jeho obnovení.

[3]Kolektor ipfixcol vyvíjený organizací CESNET https://github.com/CESNET/ipfixcol

Kapitola 3

Návrh

V této kapitole je obsažen návrh architektury a funkcí modulu. Uvedeny jsou použité nástroje, jejich výběr a návrh interpretace výstupů modulu.

Nástroj je navržen jako modul pro systém NEMEA. V obrázku 6 je zpracován diagram aktivit modulu. Aktivita zpracování dat obsahuje hlavní programovou smyčku, podrobněji zobrazenou v diagramu aktivit 7.

3.1 Popis návrhu modulu

3.1.1 Zpracování záznamů o tocích

V modulu bylo navrženo jedno vstupní síťové rozhraní, kterým jsou přijímána data ke zpracování. Přijímaná data obsahují záznamy o síťových tocích ve formátu Unirec, který je využíván v NEMEA. Zpracovávaná data ze síťových toků jsou agregována po časových intervalech o délce pěti minut. Data pro vyhodnocení jsou uchovávána v časových řadách z předchozích dvou týdnů, tedy 4032 hodnot. Interval pět minut byl zvolen z důvodu pěti minutového časového limitu pro export toků, který je v systému NEMEA výchoze nastaven. Dvou týdenní interval zpracovávaných dat je minimální doba pro použití dvou sezónní Holt-Wintersovy metody se sezónami den a týden a algoritmu pro počáteční výpočet vyhlazovacích konstant Holt-Wintersova algoritmu. Načtená data jsou zpracovávána a ukládána do grafu vytvořeného v NetworkX knihovně, popsáné v 3.5.1. V případě, že je při spuštění modulu nastaven rozsah IP adres, jsou všechny adresy mimo tento rozsah ukládány jako adresa `0.0.0.0`. K vrcholům grafu (IP adresám) a hranám (spojům mezi adresami) jsou připojena data potřebná pro vyhodnocování provozu.

3.2 Režimy použití

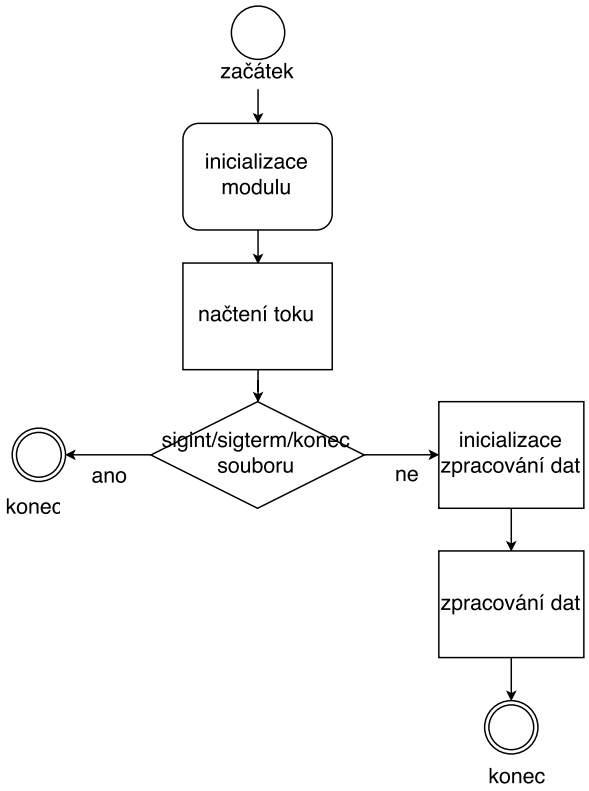
Navrženy byly dva režimy použití modulu. Režim učení a režim vyhodnocování dat. V režimu učení program načítá a zpracovává data. Po zpracování dat

z časového úseku dvou týdnů jsou naučená data uložena do souboru a modul se přepne do režimu vyhodnocování. Režim vyhodnocování může být zapnut automaticky po přepnutí z učící fáze, nebo ručně při spouštění modulu. V případě, že je spuštěn ručně, jsou načtena připravená referenční data ze souboru. Po načtení dat jsou exportovány všechny IP adresy a všechna spojení mezi IP adresami zaznamenaná v referenčních datech z učící fáze. Dále jsou uloženy soubory se známými IP adresami a známými spojeními mezi IP adresami. Jako známé jsou pro export určeny IP adresy a spojení, které se vyskytovaly ve všech časových oknech během učící fáze. Soubory se známými adresami a spojeními mezi adresami lze upravovat. Modul jejich obsah pravidelně načítá a využívá k aktualizaci seznamů známých adres a spojení během provozu. Soubory se všemi IP adresami a spojeními mohou sloužit ke kontrole výsledku učící fáze a případně je lze využít pro aktualizaci seznamů známých IP adres a známých spojení. V modulu bylo navrženo vyhodnocování dat a detekce anomálií popsané v kapitole 3.6, které s dělením na známé a neznámé adresy a spojení pracuje. Navržený výstup vyhodnocování dat je popsaný v kapitole 3.7.

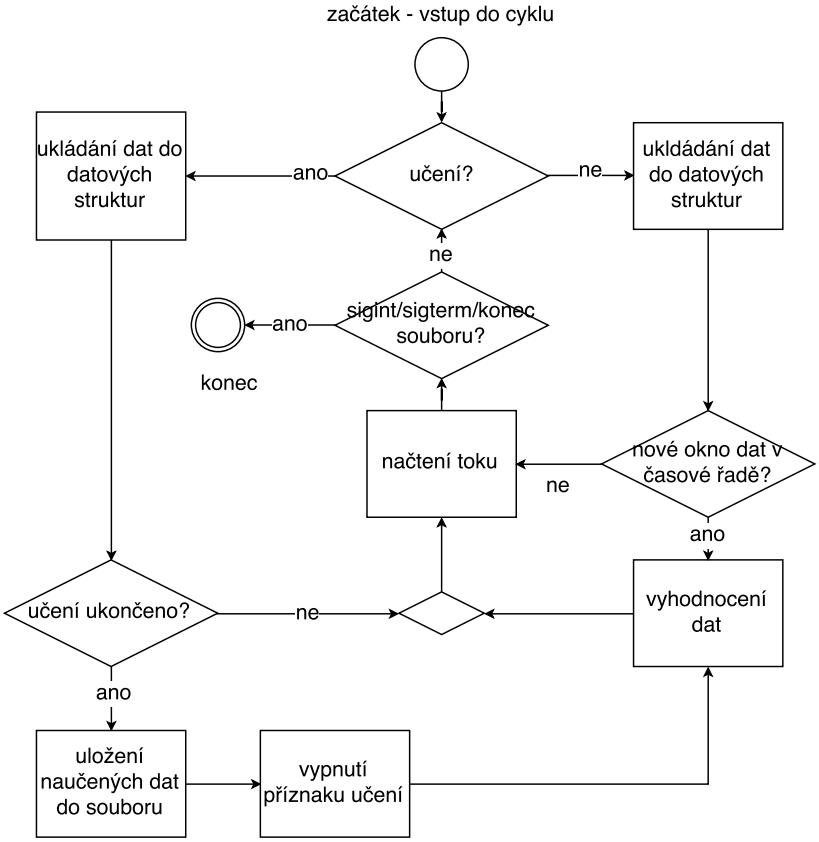
Vyhodnocené anomálie lze využít jako indikátor ukazující na možný problém. Hlášením anomálie na konkrétní IP adresy nebo spojení mezi adresami lze lépe lokalizovat problémové místo v síti.

Níže je uvedena ukázková struktury adresářů a souborů za běhu modulu v režimu vyhodnocování:

```
graphflow.py ..... hlavní skript modulu
hwt.py ..... skript s implementací Holt-Wintersovy metody
Connection-whitelist.txt ..... soubor se známými spojeními
Connection-ALL.txt ..... všechna spojení zaznamenaná při fázi učení
IP-whitelist.txt ..... soubor se známými IP adresami
IP-ALL.txt ..... soubor se všemi IP adresami z fáze učení
logs ..... adresář s logy
    graphflow-2016-04-06-14:25:54.log . soubor s textovým výstupem
    modulu
data ..... adresář obsahující soubory naučených dat
    data.json ..... soubor s daty z fáze učení
output ..... adresář s grafickými výstupy a výstupem grafu sítě
    IP-192.168.1.1 ..... adresář s grafickým výstupem pro IP adresu
        2016-04-08 03:04.png ..... graf z vyhodnocování počtu toků
    IP-192.168.1.5 ..... adresář s grafickým výstupem pro IP adresu
        2016-04-08 03:04.png ..... graf z vyhodnocování počtu toků
    Connection-192.168.1.1-192.168.1.5adresář s grafickým výstupem
    pro spojení
        2016-04-08 03:04.png ..... graf z vyhodnocování počtu toků
    total ..... adresář s vyhodnocování kompletních dat
        graph-2016-04-06 04:49.gexf ..... export grafu sítě
        2016-04-07 03:03.png ..... graf z vyhodnocování počtu toků
```



Obrázek 6: Diagram aktivit – obecný.



Obrázek 7: Diagram aktivit – zpracování dat.

3.3 Funkční požadavky modulu

Funkční požadavky na modul jsou:

- Zpracování dat ze vstupního síťového rozhraní.
- Dva režimy – režim učení a režim vyhodnocování dat.
- Data z učící fáze programu lze uložit pro pozdější spuštění modulu v režimu vyhodnocování.
- Monitorování a vyhodnocování anomálií v síťovém provozu na jednotlivých IP adresách, spojeních mezi IP adresami a v celé pozorované síti.
- Specifikace síťového rozsahu pro určení IP adres z vnitřní sítě a vnější sítě.
- Vyhodnocování připojení a odpojení IP adres, vzniku a zániku spojení mezi IP adresami.
- Ukládání informací ze síťového provozu a anomálií v grafické podobě a do logovacího souboru.
- Ukládání grafu struktury sítě.
- Funkce vyhodnocující stav provozu – stav „ve špičce“ nebo „mimo špičku“.

3.4 Zpracování grafu sítě

Při výběru nástroje pro ukládání dat grafu bylo voleno mezi nástrojem NetworkX [41] a Graph Tool [42]. V tabulce 1 jsou srovnány zásadní rozdíly mezi srovnávanými knihovnami.

V následujícím seznamu jsou vyjmenovány důvody, proč byl pro tvorbu grafů zvolen nástroj NetworkX:

- Možnost identifikace uzlu pomocí IP adresy.
- Možnost zvolit za parametry uzlů, hran a grafu libovolné hashovatelné objekty, které jsou uloženy jako list. To přináší i možnost využití kontejneru deque, který poskytuje vyšší efektivitu pro některé operace s časovými řadami než list.[43]
- Vyhledání uzlu v grafu podle identifikátoru (IP adresy) vyžaduje čas $O(1)$ díky použití hashe.

Tabulka 1: Srovnání grafových knihoven.

Networkx	Graph Tool
Implementováno v Pythonu.	Implementováno v C++.
Data jsou ukládána v hashovatelných objektech.	Data jsou ukládána do STL kontejneru vector.
Uzly grafu jsou identifikovány hashovatelným objektem.	Uzly v grafu jsou jednoznačně identifikovány indexem.
Dodatečné parametry jsou ukládány jako hashovatelné objekty a připojovány jako list k parametrům identifikujícím příslušný uzel, hranu nebo graf, ke kterému patří.	Dodatečné parametry jsou ukládány do "Property Map" třídy implementující ukládání a práci s parametry.

3.5 Použité nástroje

3.5.1 NetworkX

Pro ukládání dat ze síťových toků do grafu byla použita knihovna NetworkX. Podrobnější informace o výběru knihovny jsou uvedeny v kapitole 3.4. Jedná se o softwarový balík vhodný pro vytváření, manipulaci a studování struktury grafů. Některé základní vlastnosti uváděné v [44] jsou:

- Zpracování dat založené na datových strukturách Pythonu.
- Obsahuje mnoho grafových algoritmů.
- Obsahuje nástroje pro měření a analýzu grafů.
- Umí generovat různé druhy grafů (klasický, náhodný,...).
- Možnost volby identifikátoru uzlu (např. text, obrázek, xml záznam).
- Hrany mohou obsahovat různá data (např. váhu, časové řady).
- Otevřený zdrojový kód.

Knihovna umožňuje kromě volání složitějších algoritmů nad grafem jednoduchou správu grafu. Níže je uveden příklad použití:

```
import networkx as nx
#Vytvoření instance grafu.
G=nx.Graph()
#přidání uzlů se názvem podle IP adresy
G.add_node("192.168.1.1")
G.add_node("192.168.1.2")
```

```
#Přidání hrany k~uzlům.
G.add_edge("192.168.1.1","192.168.1.2")
#Odebrání uzlu odstraní i případné hrany
vedoucí mezi odebíraným uzlem a ostatními uzly.
G.remove_node("192.168.1.1")
```

3.5.2 Matplotlib

K tvorbě grafů byla využita opensource knihovna `Matplotlib` [45].
Tato knihovna umožňuje vytvářet různé druhy `2D` grafů s širokými možnostmi nastavení. Knihovna umožňuje importování do Pythonu a využití v dalších nástrojích (Mathematica, Matlab). Pro různé případy užití umožňuje nastavit různé druhy backendu[4], které používají k vytvoření grafu různé technologie s odlišnými funkcemi a rychlostí. Níže je uvedena ukázka použití knihovny k vytvoření a uložení grafu do souboru:

```
#vytvoření instance grafu o~zadané velikosti (v~palcích)
plt.figure(None, (7,5))
#nastavení fontu
plt.rc('font', family='serif', size=14)
#nastavení pozice popisku
plt.legend(loc='upper center', bbox_to_anchor=(0.5, 1.15),
fancybox=True, shadow=True, ncol=5)
#uložení grafu do souboru
plt.savefig(str(img_path) + '.png')
```

3.5.3 Záznam událostí

Pro textový záznam výstupu modulu je využita knihovna `logging`. Tato knihovna umožňuje flexibilní záznam událostí a nabízí široké možnosti přizpůsobení. Záznam událostí je tříděn podle úrovní důležitosti. Podle zvolené úrovně je zaznamenávána jen odpovídající část událostí. Každá úroveň má textovou a číselnou hodnotu. Číselná hodnota od nejnižšího po nejvyšší určuje důležitost. Knihovna umožňuje kompletní přizpůsobení obsahu a zaznamenávaných událostí a jejich úrovní. V ukázce zdrojového kódu je vidět základní nastavení pro logování do souboru a využití logování na úrovní debug, info a warning.

```
import logging
logging.basicConfig(level=logging.DEBUG,
format='%(asctime)s %(levelname)-8s %(message)s',
datefmt='%a, %d %b %Y %H:%M:%S', filename='/temp/myapp.log',
```

[4] Část aplikace, která se stará o základní funkce programu.

```
filemode='w')
logging.debug('Information message.')
logging.info('Debug message.')
logging.warning('Warning message.')
```

Výstupem v souboru `myapp.log` bude:

```
Fri, 02 Jul 2004 13:06:18 DEBUG    Information message.
Fri, 02 Jul 2004 13:06:18 INFO     Debug message.
Fri, 02 Jul 2004 13:06:18 WARNING  Warning message.
```

3.6 Monitorování a detekce anomálií

3.6.1 Detekování anomálií v počtu síťových toků

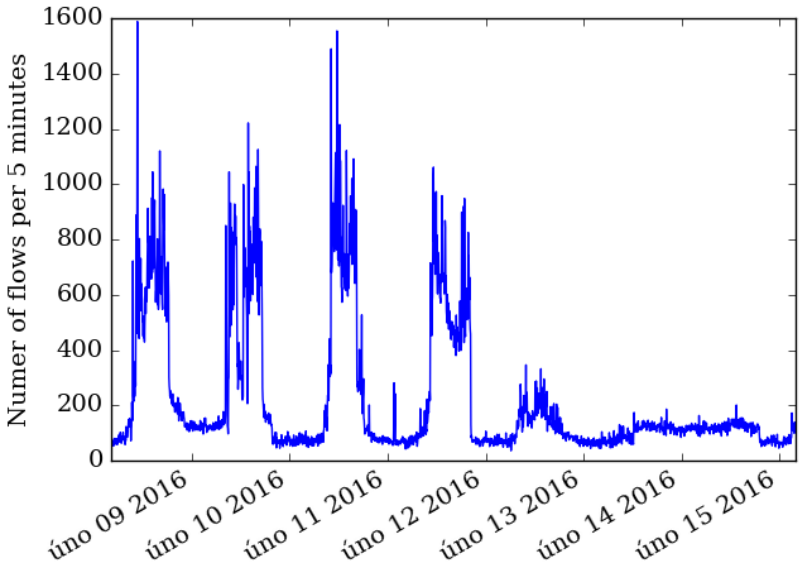
Modul umožňuje detekovat anomálie v provozu založené na množství síťových toků. Vyhodnocování provozu modul provádí pro jednotlivé připojené IP adresy, spojení mezi IP adresami a celý monitorovaný provoz. Detekční metoda zpracovává data z časových řad. Pro zpracování časových řad je využita Holt-Wintersova metoda ve verzi se dvěma sezónami popsaná v kapitole 2.6.7. Jedna sezóna je nastavena na dobu jednoho dne, druhá sezóna je nastavena na dobu jednoho týdne. Tímto nastavením jsou pokryty výkyvy v datech, které způsobují rozdíly v používání sítě během pracovních dnů a víkendu. Obrázek 8 ukazuje záznam provozu zachyceného v testovací síti u vedoucího práce během jednoho týdne. V obrázku je vidět sezónnost ve dnech a v celém týdnu, která se opakovala během několikatýdenního pozorování. Detekce anomálií s využitím předpovědi následující hodnoty Holt-Wintersovou metodou byla navrhnuta podle kapitoly 2.4.1.

Vyhodnocování anomálií v počtu toků probíhá pouze u IP adres a spojů mezi IP adresami, které jsou v síti aktivní po celou dobu dvou týdnů ve všech časových oknech.

Nasazení Holt-Wintersovy metody je v systému navrženo tak, aby před počátkem detekce byly automaticky nastaveny všechny vyhlazovací parametry metody. Parametry jsou určovány pomocí minimalizační funkce `fmin_l_bfgs_b` [46] ze statistické knihovny `SciPy` [47]. Tato funkce využívá pro výpočet trénovací a testovací sérii dat. Každá série má velikost delší ze dvou počítaných period (týdenní). Metoda proto potřebuje před připravená dvoutýdenní referenční data.

3.6.2 Detekování změn ve struktuře sítě

Modul umožňuje detekci změn ve struktuře sítě na základě kontroly IP adres. Detekované změny jsou rozděleny na úrovně podle důležitosti. Detekované změny jsou podle úrovně důležitosti vyjmenovány v kapitole 3.7.1.



Obrázek 8: Ukázka periodického chování sítě.

V této kapitole jsou popsány způsoby určení změn ve struktuře sítě. V textu se vyskytuje pojem známá IP adresa a známé spojení mezi IP adresami. Adresa nebo spojení jsou určeny za známé, pokud se nachází v seznamu známých adres nebo seznamu známých spojení. Tyto seznamy jsou uloženy v textových souborech. Modul za chodu obnovuje své interní seznamy známých adres pravidelným čtením těchto souborů.

Připojení IP adresy je detekováno, pokud je v aktuálně vyhodnocovaném časovém okně adresa zaznamenána a v předešlém časovém okně zaznamenána nebyla.

Odpojení IP adresy je detekováno, pokud není v aktuálně vyhodnocovaném časovém okně adresa zaznamenána a v předešlém časovém okně zaznamenána byla.

Začátek komunikace mezi dvěma IP adresami je detekován, pokud je v aktuálně vyhodnocovaném okně zaznamenáno spojení mezi zdrojovou a cílovou IP adresou, které v minulém časovém okně zaznamenáno nebylo.

Zaniknutí komunikace mezi dvěma IP adresami je detekováno, pokud není v aktuálně vyhodnocovaném časovém okně zaznamenáno spojení mezi zdrojovou a cílovou IP adresou, které v minulém časovém okně zaznamenáno bylo.

Tyto změny modul detekuje zvlášť pro známé a neznámé IP adresy a spojení. Změna stavu provozu označeného jako „ve špičce“ a „mimo špičku“ je detekována pomocí prahové hodnoty počtu toků, kdy provoz pod danou hranicí je považován za „mimo špičku“ a nad danou hranicí „ve špičce“. Tato

hranice je volena při spuštění modulu. Změna je detekována, pokud se po určený časový interval provoz nachází v jednom ze stavů „ve špičce“ nebo „mimo špičku“.

3.7 Výstup vyhodnocování

3.7.1 Textový výstup

Pro textový výstup modulu je využita knihovna Logging. Výstupy jsou rozděleny do dvou úrovní: **info** a **anomaly**. Při spouštění modulu lze zvolit, jestli má být zaznamenávána pouze úroveň anomaly nebo obsáhlejší detekce info i anomaly.

Úroveň info zaznamenává velké množství informací pro poskytnutí co nejpodrobnějšího dohledu nad sítí:

- Připojení IP adresy známé z fáze učení.
- Odpojení IP adresy známé z fáze učení.
- Začátek komunikace mezi dvěma IP adresami, známé z fáze učení.
- Zaniknutí komunikace mezi dvěma IP adresami, známé z fáze učení.
- Informace, že existuje IP adresa, která nově splňuje podmínku, že byla aktivní ve všech oknech časové řady, a bude nově vyhodnocována na anomálie Holt-Wintersovou metodou.
- Informace, že existuje spojení mezi IP adresami, které nově splňuje podmínku, že bylo aktivní ve všech oknech časové řady, a bude nově vyhodnocováno na anomálie Holt-Wintersovou metodou.
- Odpojení IP adresy, která byla aktivní ve všech časových oknech zpracovávané dvou týdenní časové řady (byla vyhodnocována na anomálie Holt-Wintersovou metodou).
- Zaniknutí komunikace mezi IP adresami, která byla aktivní ve všech časových oknech zpracovávané dvou týdenní časové řady (byla vyhodnocována na anomálie Holt-Wintersovou metodou).
- Změna stavu provozu označeného jako „ve špičce“ a „mimo špičku“.

Úrovní anomaly jsou zaznamenávány události:

- Anomálie v množství síťových toků za časové okno na IP adresu.
- Anomálie v množství síťových toků za časové okno ve spoji ze zdrojové do cílové IP adresy.

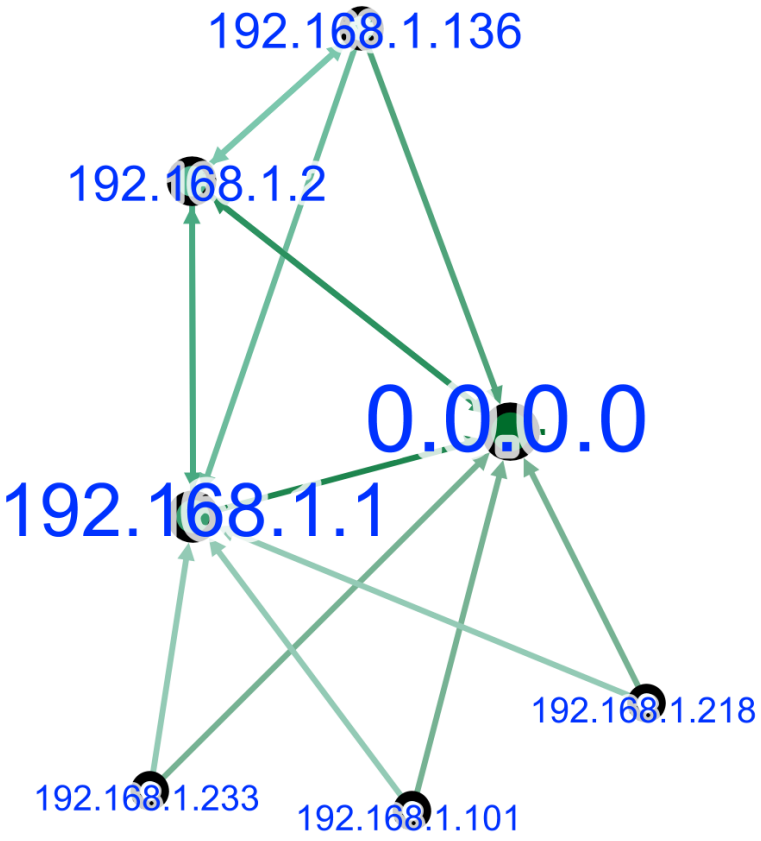
- Anomálie v celkovém množství síťových toků za časové okno ve sledované síti.
- Začátek komunikace mezi dvěma IP adresami, neznámé z fáze učení.
- Ukončení komunikace mezi dvěma IP adresami, neznámé z fáze učení.
- Připojení IP adresy neznámé z fáze učení.
- Odpojení IP adresy neznámé z fáze učení.

3.7.2 Export grafů

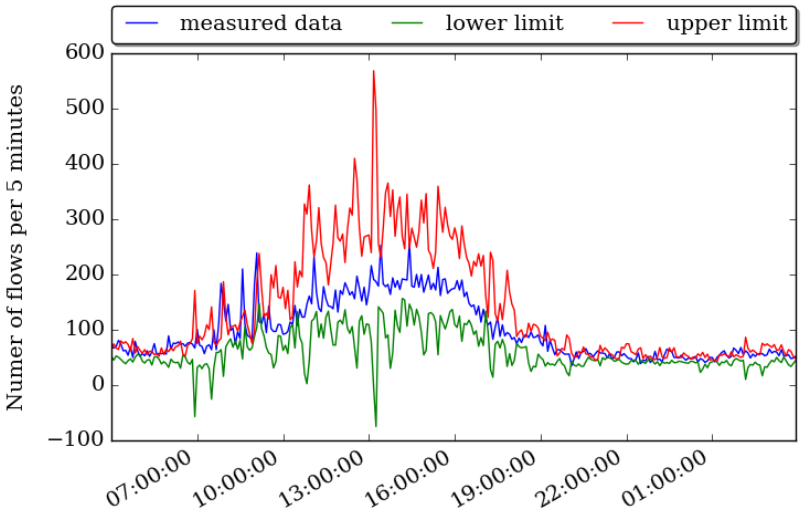
Pro IP adresy a spoje mezi IP adresami, které jsou zpracovávány Holt-Wintersovou metodou a pro celou pozorovanou síť, exportuje modul grafy zobrazující naměřený síťový provoz a prahové hodnoty detekce anomálie odvozené z chyby předpovědi Holt-Wintersovy metody. Interval tvorby grafů a cestu pro export lze uživatelsky volit při spouštění modulu.

Pro kontrolu struktury sítě je s grafy provozu exportován i graf struktury sítě, který zobrazuje IP adresy a spoje mezi IP adresami, které se uskutečnily po dobu nastaveného intervalu mezi exportem grafů. Struktura je exportována do formátu `gexf`. Na obrázku 9 je vidět ukázka grafu struktury testovací sítě u vedoucího práce dne 26. 1. 2016. Vizualizace pochází z nástroje `Gephi` [48].

V obrázku 10 je zachycen standardní průběh celkového provozu během jednoho dne v testovací síti. Vidět jsou naměřená data a horní a dolní prahové hodnoty pro vyhodnocení anomálie v počtu toků.



Obrázek 9: Ukázka grafu struktury sítě.

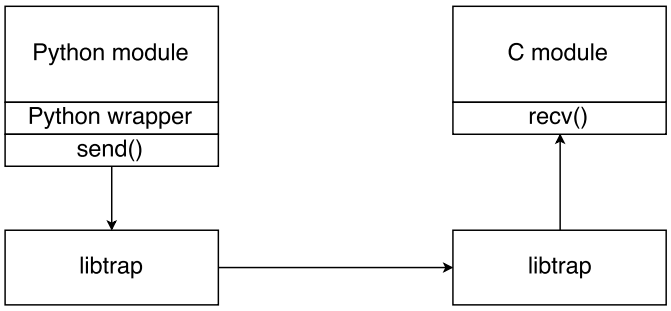


Obrázek 10: Ukázka grafického výstupu z Holt-Wintersovy metody.

Kapitola 4

Nasazení modulu v NEMEA

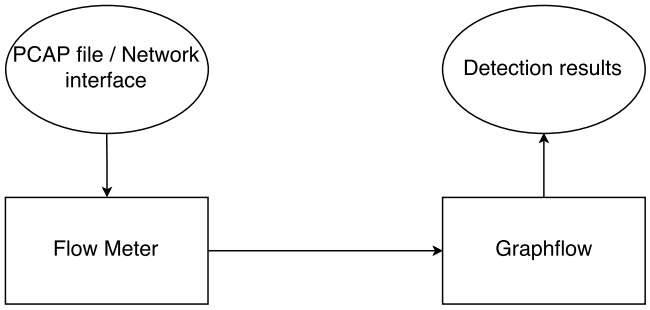
NEMEA poskytuje framework umožňující tvorbu modulů, které implementují různé síťové nástroje. V NEMEA je příprava pro využití jazyků C/C++ a Python. Pro spojení mezi moduly je využita platforma TRAP ve sdílené knihovně libtrap. Pro komunikaci je využit formát dat Unirec. V případě implementace v Pythonu je k dispozici Python wrapper, obalující knihovnu libtrap a Python knihovna Unirec implementující Unirec formát v Pythonu. Na obrázku 11 je vidět spojení Python modulu a modulu v jayzce C.



Obrázek 11: Napojení modulů v NEMEA.

Navrhovaný modul poskytuje jedno vstupní rozhraní, které lze napojit do systému NEMEA. Díky variabilitě modulárního systému NEMEA lze připojit modul více způsoby v závislosti na prostředí, kde je spouštěn. Síťové toky ve formátu Unirec může modul přijímat od jiných modulů v NEMEA, které je poskytují na výstupním rozhraní. Obrázek 12 ukazuje minimální zapojení v prostředí, kde je využit modul Flow meter, který na vstupním rozhraní při-

jímá data ze síťového rozhraní nebo PCAP[5] souboru. Přijatá data zpracovává na toky, které odesílá na výstupním rozhraní ve formátu Unirec. Na výstupní rozhraní Flow meteru je v obrázku připojen modul modul, který přijímá a zpracovává Unirec data.



Obrázek 12: Příklad nasazení modulu.

[5]Rozhraní pro zachytávání dat ze sítě.

Kapitola 5

Implementace

5.1 Struktura navrhnutého modulu

Nástroj je implementován v jazyce Python [49] verze 2.7.11 a skládá se ze dvou skriptů:

- `graphflow.py`, hlavní skript, který provádí zpracování a vyhodnocení dat.
- `hwt.py`, skript implementující Holt-Wintersovu statistickou metodu, zpracovávající časové řady a předpověď hodnoty v následujícím časovém okně pro využití v hlavním skriptu graphflow.py.

V následujícím seznamu jsou vypsány hlavní funkce, které byly implementovány ve skriptu graphflow.py, společně s jejich stručným popisem:

- `FlowProcess`: Funkce zpracovávající hlavní smyčku v programu. Volá funkce pro načítání, zpracování a ukládání dat.
- `DataLoader`: Funkce implementující napojení na Python wrapper NEMEA systému a získávání dat pro zpracování.
- `FillGraph`: Funkce řídící zpracování načtených dat, tvorbu datových struktur, časových řad.
- `EdgeAnalysis`: Funkce zpracovávající vyhodnocení a detekci anomálií v provozu ve spojích mezi IP.
- `NodeAnalysis`: Funkce zpracovávající vyhodnocení a detekci anomálií v provozu na jednotlivé IP.
- `TotalFlowAnalysis`: Funkce zpracovávající vyhodnocení a detekci anomálií z celkového provozu.
- `PlotData`: Funkce řídící tvorbu grafů provozu v síti a struktury sítě.

- `QuietPeriodProcess`: Funkce pro detekování provozu „ve špičce“ a „mimo špičku“.
- `ExportData`: Funkce pro export dat z učícího období do souboru.
- `ImportData`: Funkce pro import připravených dat z fáze učení ze souboru.

V souboru hwt.py jsou funkce pro výpočet předpovědi z časové řady Holt-Wintersovou metodou:

- `HWT`: Hlavní funkce spouštěná při prvním volání Holt-Wintersovy. Funkce se stará o výpočet vyhlazovacích parametrů pomocí minimalizační funkce `fmin_l_bfgs_b`, zpracování naučených dat a vytvoření první predikce.
- `HWTStep`: Funkce počítající jednu iteraci Holt-Wintersovy metody.
- `RMSE`: Funkce pro výpočet střední kvadratické chyby. Je využívána při prvotním výpočtu vyhlazovacích statistik Holt-Wintersovy metody.
- `ParamsEstimation`: Funkce využívaná pro výpočet Holt-Wintersovy metody ve funkci RMSE.

5.2 Načítání dat

Modul načítá data ve formátu Unirec používaném v NEMEA. Pro komunikaci Unirec dat přes libtrap popsanou v kapitole 4 je potřeba provést import python wrapperu trap a knihovny unirec. Po splnění těchto podmínek může proběhnout inicializace:

```
#sestavení specifikace rozhraní ze zadaných parametrů modulu
ifc_spec = trap.parseParams(sys.argv, module_info)
#inicializace libtrap knihovny
trap.init(module_info, ifc_spec)
#zaregistrování signálů SIGINT a DIGTERM pro ukončení modulu
trap.registerDefaultSignalHandler()
# nastavení požadovaného formátu akceptovaného modulem
#(zde povolena všechna políčka Unirec formátu)
trap.set_required_fmt(0, trap.TRAP_FMT_UNIREC, "")
```

Po inicializaci lze přijímat a odesílat data přes nastavená rozhraní. NEMEA umožňuje dynamickou změnu formátu dat během provozu modulu, proto je při načítání dat ošetřován i tento případ. Načítání dat v modulu poté vypadá takto:

```
try:
    #příjem dat přes libtrap
    data = trap.recv(0)
#výjimka nastane, pokud inicializované rozraní nebo
#formát dat neodpovídá
except trap.EFMTMismatch:
    return -1, -1
#výjimka nastane, pokud je vyvolána změna formátu dat,
#poté jsou nastaveny potřebné struktury podle nového formátu
except trap.EFMTChanged as e:
    (fmttype, fmtspec) = trap.get_data_fmt(trap.IFC_INPUT, 0)
    UR_Flow = unirec.CreateTemplate("UR_Flow", fmtspec)
    data = e.data
#jiná výjimka v~datech
except trap.ETerminated:
    return -1, -1
#pokud je velikost načtených dat < 1, je ukončeno načítání
if len(data) <= 1:
    return -1, -1
#vrácení načtených Unirec dat a formátu dat
return UR_Flow(data), UR_Flow
```

5.3 Zpracování dat

V této kapitole je popsána implementace hlavního programového cyklu zpracování dat, rozdělena podle dvou režimů, ve kterých modul může pracovat. Režimu učení a vyhodnocování dat.

Zpracování dat je řízeno z hlavní funkce `FlowProcess`. V případě, že je spuštěn režim učení, je vynecháno volání všech vyhodnocovacích funkcí. V režimu vyhodnocování jsou vyhodnocovací funkce využity.

5.3.1 Režim učení

Hlavní programová smyčka ve `FlowProcess` spouští načítání dat funkcí `DataLoader` a zpracování do grafu funkcí `FillGraph`. Funkce `FillGraph` implementuje zpracování dat přes vyrovnávací paměť a volá funkci `AddRecord`. Přidání záznamu do grafové struktury řídí funkce `AddRecord`, která volá funkce pro zpracování a ukládání uzlů a hran do grafu a dalších dat potřebných pro vyhodnocování:

```
src_ip = CheckIPRange(str(rec.SRC_IP), ip_range)
dst_ip = CheckIPRange(str(rec.DST_IP), ip_range)
gr = FillGraphRecord(gr, rec, next_period)
gr = NextPeriodProcess(gr, next_period)
```

```
gr = FillNodeRecord( src_ip, rec, gr )
gr = FillNodeRecord( dst_ip, rec, gr )
gr = FillEdgeRecord( src_ip, dst_ip, rec, gr )
```

Zpracováním dat je vytvářena grafová struktura s daty potřebnými ve vyhodnocovací fázi. V případě, že program nasbírá dostatek dat pro vyhodnocování, jsou předzpracovaná data uložena do souboru funkcí `ExportData` a program se přepíná do vyhodnocovací fáze.

5.3.2 Režim vyhodnocování dat

Do režimu vyhodnocování dat se může program dostat dvěma způsoby:

- Přepnutím z režimu učení. Modul je po nasbírání dostatku dat automaticky přepnut do vyhodnocovacího režimu.
- Spuštěním v režimu vyhodnocování. Modul při spuštění načte funkcí `ImportData` soubor s naučenými daty pro běh vyhodnocovacího algoritmu.

Při inicializaci režimu vyhodnocování jsou z naučených dat exportovány soubory se všemi IP adresami a spojeními mezi IP adresami zaznamenanými během fáze učení. Dále je exportován soubor obsahující IP adresy a soubor obsahující spojení mezi IP adresami které jsou považovány za známé. Za známé jsou považovány adresy a spojení, které byly aktivní ve všech časových oknech ve fázi učení. V režimu vyhodnocování jsou data načítána a zpracována stejným způsobem jako v režimu učení. Navíc jsou volány funkce vyhodnocování, udržování a čištění grafu, synchronizace známých IP adres a spojení se soubory známých IP adres a spojení:

```
CleanGraph(gr)
known_nodes_set = ImportWhitelist(whitelist_file_path,"node")
known_edges_set = ImportWhitelist(whitelist_file_path,"edge")
TotalFlowAnalysis(gr,  NUM_PERIODS_REPORT)
NodeAnalysis(gr,  NUM_PERIODS_REPORT,known_nodes_set)
EdgeAnalysis(gr,  NUM_PERIODS_REPORT,known_edges_set)
quiet_period = QuietPeriodProcess(gr, THRESHOLD,
TRAFFIC_STATE_CHANGE_PERIOD,quiet_period)
```

Tyto funkce jsou v režimu vyhodnocování volány vždy, když je vytvořeno nové okno časové řady se zpracovávanými daty. Pro tvorbu a ukládání grafů je volána funkce `PlotData`.

Funkce `CleanGraph` má na starost pročišťování grafu od trvale neaktivních IP adres, spojů mezi IP adresami a odstraňování s nimi souvisejících nepotřebných datových struktur.

Funkce `TotalFlowAnalysis` má na starosti monitorování a detekci anomálií z kompletního zachytávaného provozu. V této funkci jsou volány funkce pro získání predikce dalšího rámce časové řady, výpočet prahových hodnot pro detekci a detekci anomálií v počtu síťových toků. V případě, že je vyhodnocena anomálie, jsou data pro výpočet Holt-Wintersovy metody nahrazována daty vypočítanými z předpovědi, místo dat vypočítaných z reálného provozu během anomálie. Tímto způsobem se snižuje riziko budoucí zkreslené předpovědi naučením dat během anomálie.

V `TotalFlowAnalysis` je volána funkce pro výpočet prahových hodnot detekce. Vzorec pro výpočet založený na exponenciálním vyhlazování vyžaduje data z předpovědi o stáří jedné týdenní sezóny, která nejsou první týden vyhodnocování dostupná, z tohoto důvodu jsou během prvního týdne data brána z předchozí iterace předpovědi jako absolutní hodnota rozdílu naměřené hodnoty posledního časového rámce a předpovězené hodnoty pro poslední časový rámec:

```
gr.graph['hwt_deviation'].append(abs( gr.graph['flow_count'][-1]
  -  gr.graph['hwt_flow'][-1]))
```

Po získání dostatečného množství dat je výpočet konfidenčního intervalu již podle vzorce uvedeného v kapitole 2.4.1 :

```
gr.graph['hwt_deviation'].append(abs((gr.graph['hwt_params'][1]
  * ( gr.graph['flow_count'][-1] -  gr.graph['hwt_flow'][-1]))
  + ((1 -  gr.graph['hwt_params'][1])
  * gr.graph['hwt_deviation'][-WEEK_AGGREGATION_PERIODS_COUNT])))
```

Funkce `NodeAnalysis` vyhodnocuje anomálie v počtech síťových toků obdobným způsobem jako funkce `TotalFlowAnalysis`. Vyhodnocování probíhá pro všechny IP adresy ve sledované síti, které byly aktivní ve všech oknech časové řady vyhodnocovaných dat. Funkce `NodeAnalysis` dále vyhodnocuje změny ve struktuře sítě popsané v návrhu 3.6.2.

Funkce `EdgeAnalysis` vyhodnocuje změny ve struktuře sítě a počtu síťových toků obdobným způsobem jako funkce `NodeAnalysis`. Vyhodnocuje změny ve struktuře sítě na základě vznikání a zanikání spojení mezi zdrojovou a cílovou IP adresou. Funkce dále vyhodnocuje anomálie v počtech toků na jednotlivých spojích mezi zdrojovou a cílovou IP adresou. Vyhodnocovány jsou anomálie na všech spojích, které byly aktivní ve všech oknech vyhodnocované časové řady.

Funkce `QuietPeriodProcess` je funkce vyhodnocující změny stavu provozu „ve špičce“ a „mimo špičku“. Vyhodnocování probíhá na základě předem zvolené konstanty.

Funkce `PlotData` vytváří a exportuje grafy z provozu sítě a grafy struktury sítě.

Kapitola 6

Testování

6.1 Testovací prostředí

Testování probíhalo ve dvou prostředích. Vyhodnocování v reálném čase probíhalo ve virtuálním stroji s instalovaným operačním systémem Fedora 19, procesorem 3,1GHz a 1GB operační paměti. Vyhodnocována byla data ze sítě v malé firmě zaměřující se na IT zakázky. Testování z předem uložených záznamů o tocích probíhalo na notebooku s procesorem Intel Core i5 s maximální frekvencí 2,7 GHz, 8GB operační paměti a nainstalovaným operačním systémem Debian 8.

V testovací síti byl zapojen wifi směřovač a PC s běžícím systém NEMEA. Dále byla do směřovače průběžně zapojována a odpojována další zařízení (notebooky, telefony). Provoz v síti byl směřován přes PC s NEMEA, který zpracovával a exportoval záznamy o síťových tocích. Ve firemní síti byl ve virtuálním prostředí nasazen obraz operačního systému Fedora verze 19 obsahující instalaci NEMEA. Obraz pro testování poskytl vedoucí práce. Do systému byl doinstalován modul vyvíjený v této práci. Na síťové rozhraní testovacího virtuálního stroje byl přesměrován zrcadlený provoz z několika firemních síťových rozsahů. Experimenty probíhaly na těchto síťových rozsazích:

- Rozsah vyhrazený firemním serverům.
- Rozsah vyhrazený zařízením zaměstnanců ve vnitřní síti.

Na testovacích datech získaných od vedoucího práce bylo provedeno testování využití operační paměti nástrojem Massif. Graf využití paměti je uveden v příloze D.

6.2 Výsledky testování

6.2.1 Vyhodnocení anomálií v počtu toků

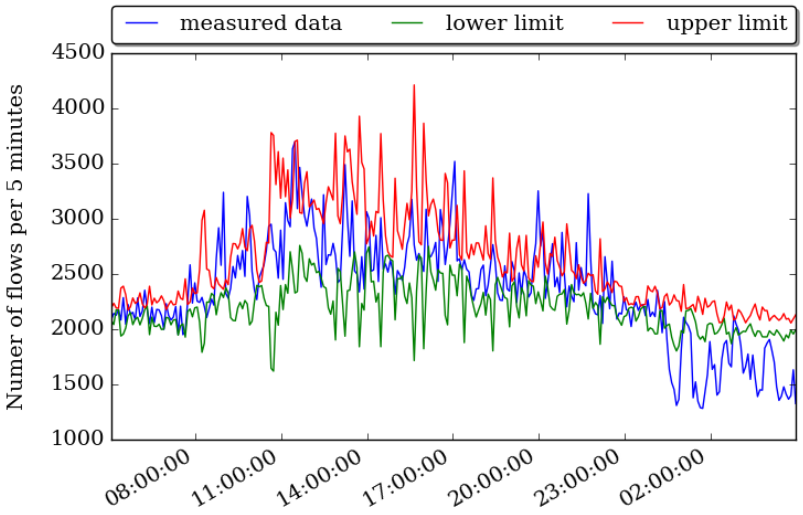
Všechna vyhodnocování probíhala s nastavením konstant δ_- a δ_+ pro úpravu prahových hodnot pro detekci na hodnotu $2,5$. Hodnota byla zvolena na základě článku [50], kde jsou doporučeny volit hodnoty mezi $2-3$. Pro snížení množství falešně pozitivních detekcí byla podle 2.4.1 nastavena detekce anomálie při vyhodnocení dvou časových oken s naměřenými hodnotami překračujícími prahové hodnoty pro detekci.

První testování ve firemním prostředí probíhalo na rozsahu adres pro servery.

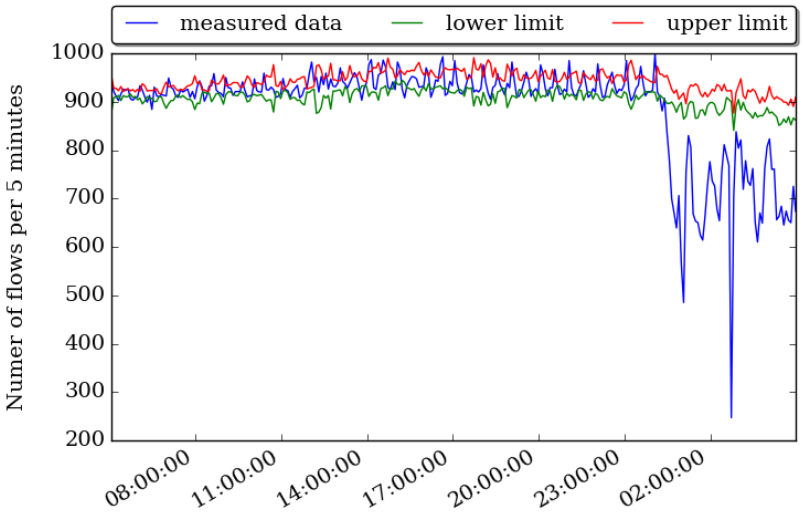
Ve firemní síti byla v noci z 11.4. na 12.4. zaznamenána anomálie na více IP adresách, spojeních i ve vyhodnocování za celý pozorovaný síťový rozsah. Anomálie se projevila jako pokles počtu toků, který začal podle záznamu v logu v 22:28 UTC času. Při další analýze bylo zjištěno, že anomálie byla hlášena na spojeních mezi IP ze sledovaného rozsahu s IP mimo sledovaný síťový rozsah a to v obou směrech. Na sledovaných spojeních pouze v rámci vyhodnocovaného rozsahu anomálie hlášena nebyla. Dalším zjištěním bylo, že anomálie byla vyhodnocena pouze na některých ze serverů komunikujících mimo specifikovaný síťový rozsah firemních serverů a to těch, které jsou napojené na servery v cloudu, ve kterém je umístěna část firemní serverové infrastruktury. Z těchto zjištění bylo usouzeno, že by mohl být problém s VPN připojením na cloud, nebo jiný technický problém v cloudu. Problém byl objasněn tím, že v této době probíhal noční přenos většího množství dat přes VPN z cloudu do sítě, která nebyla modulem monitorována. To způsobilo aktuální snížení propustnosti přes VPN pro zařízení ze sledovaného rozsahu, které se projevilo snížením počtu toků z některých zařízení a spojení ve sledovaném síťovém rozsahu.

V obrázku 13 je vidět zaznamenaná anomálie v počtu toků ve spojení z oblasti mimo sledovaný rozsah adres do jednoho ze serverů ve sledovaném rozsahu. Na obrázku 14 je zachycena anomálie ve stejné době v komunikaci z oblasti mimo sledovaný rozsah do jiného serveru ve sledovaném rozsahu. Hodnoty se vrátily zpět do předpovídaného intervalu druhý den ráno.

Ve firemní síti byly zaznamenána další hlášení, která byla dodatečně vyhodnocena jako falešně pozitivní, na hlášených zařízeních nebyla zaznamenána žádná nežádoucí aktivita, záznamy v logách, ani přítomnost škodlivého softwaru. Množství těchto hlášení se lišilo v jednotlivých vyhodnocovaných zařízeních a spojeních mezi zařízeními. Časté hlášení anomálií bylo zaznamenáno na zařízeních a spojeních s většími nepravidelnými změnami v množství provozu.



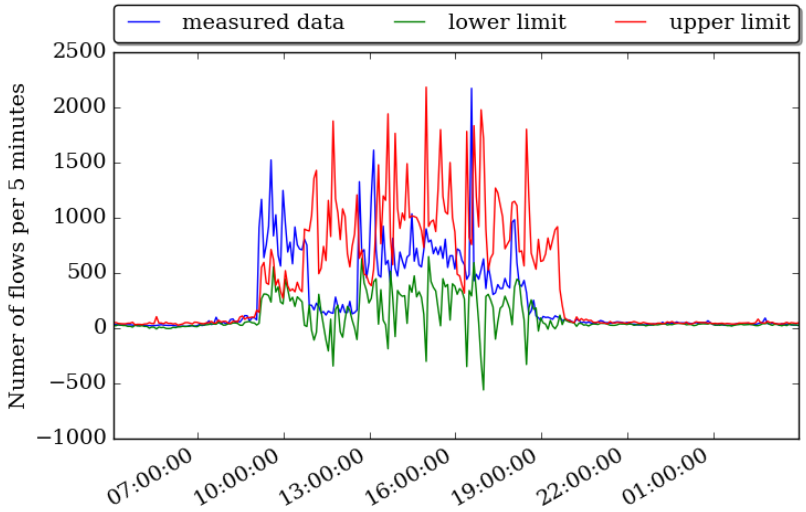
Obrázek 13: Vyhodnocení anomálie ve fremní síti 11. 4. případ 1.



Obrázek 14: Vyhodnocení anomálie ve fremní síti 11. 4. případ 2.

Ze sítě u vedoucího práce byla zpracována data od 17. 11. 2015 do 21. 12. 2015, Vyhodnocování anomálií bylo provedeno v době od 24. 11. do 21. 12. Z dat v týdnu 17. 11 až 24. 11. byl inicializován výpočet prahových hodnot pro vyhodnocení anomálie uvedený v kapitole 2.4.1. Ve vyhodnocovaném intervalu byly hlášeny dvě uměle vyvolané anomálie.

Dne 30. 11. proběhl brute-force útok, dále byl provoz ovlivněn stahováním a instalací OS Fedora. Tato anomálie je viditelná v obrázku 15 jako nárůst počtu toků nad prahovou hodnotu v dopoledních hodinách. Následující pokles v počtu toků okolo poledne k dolní hranici pro detekci by mohl znamenat ukončení experimentu a přestávku. Během dne byly zaznamenány další tři krátkodobé zvýšení počtu toků nad prahovou hodnotu.



Obrázek 15: Vyhodnocení anomálie 30. 11.

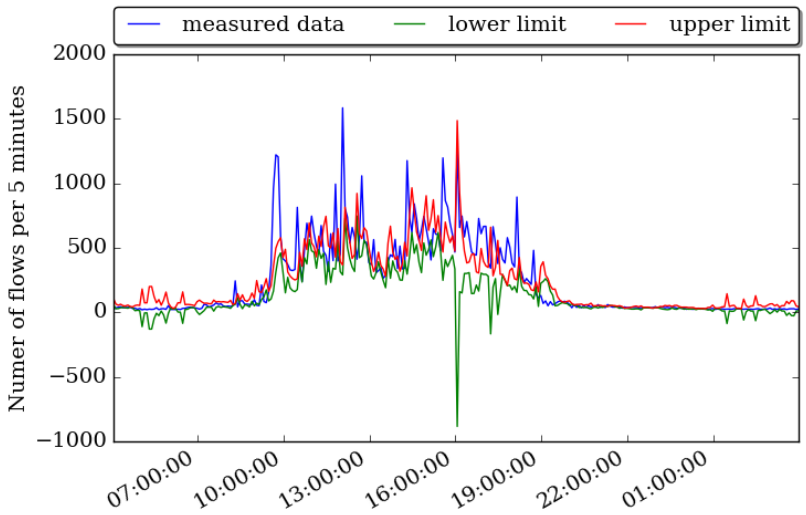
Anomálie v počtu toků v dopoledních hodinách 30. 11. byla v hlášeních v celkovém sledovaném provozu takto:

```
2016-04-19 19:41:54,017 - ANOMALY - Average flow count: 1067,
 average prediction count: 393 per 5 miutes during last
 15 minutes before 2015-11-30 08:33-UTC
2016-04-19 19:41:55,322 - ANOMALY - Average flow count: 840,
 average prediction count: 452 per 5 miutes during last
 15 minutes before 2015-11-30 08:48-UTC
2016-04-19 19:41:56,448 - ANOMALY - Average flow count: 918,
 average prediction count: 338 per 5 miutes during last
 15 minutes before 2015-11-30 09:03-UTC
2016-04-19 19:41:57,372 - ANOMALY - Average flow count: 684,
 average prediction count: 326 per 5 miutes during last
 15 minutes before 2015-11-30 09:18-UTC
2016-04-19 19:41:58,747 - ANOMALY - Average flow count: 796,
 average prediction count: 307 per 5 miutes during last
 15 minutes before 2015-11-30 09:33-UTC
2016-04-19 19:42:04,327 - ANOMALY - Average flow count: 1132,
 average prediction count: 374 per 5 miutes during last
 15 minutes before 2015-11-30 12:09-UTC
```

Druhá uměle vyvolaná anomálie byl DOS útok 9. 12. 2015. V grafu celkového provozu v síti z tohoto dne 16 jsou před desátou hodinou dopoledne a okolo poledne vidět dva výkyvy, které ukazují zvýšení počtu toků. Dále je vidět detekovaný vyšší provoz i během odpoledních hodin.

V průběhu dopolední anomálie byly hlášeny v logovacím souboru následující záznamy o anomáliích pro celkový sledovaný provoz:

```
2016-04-19 21:05:27,756 - ANOMALY - Average flow count: 198,
 average prediction count: 124 per 5 miutes during last
 15 minutes before 2015-12-09 08:33-UTC
2016-04-19 21:05:29,123 - ANOMALY - Average flow count: 1128,
 average prediction count: 345 per 5 miutes during last
 15 minutes before 2015-12-09 08:48-UTC
2016-04-19 22:27:37,525 - ANOMALY - Average flow count: 614,
 average prediction count: 386 per 5 miutes during last
 15 minutes before 2015-12-09 10:58-UTC
2016-04-19 22:27:38,750 - ANOMALY - Average flow count: 1043,
 average prediction count: 465 per 5 miutes during last
 15 minutes before 2015-12-09 11:14-UTC
2016-04-19 22:27:39,635 - ANOMALY - Average flow count: 550,
 average prediction count: 501 per 5 miutes during last
 15 minutes before 2015-12-09 11:33-UTC
```



Obrázek 16: Vyhodnocení anomálie 9. 12.

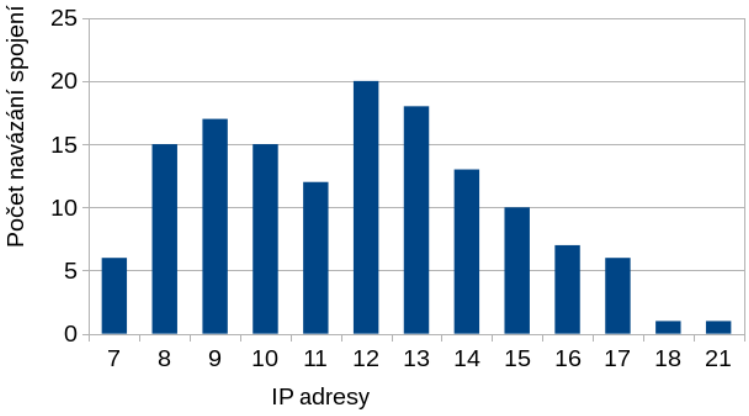
Některé z detekcí mohou být způsobeny tím, že byla data sbírána z malého množství síťového provozu, vytvářeného používáním sítě řádově jednotkami lidí. Provoz v síti je proto nepravidelný s velkými výkyvy. Z tohoto důvodu byly odhaleny v síti další anomálie v počtu toků i v jiných časech.

6.2.2 Vyhodnocení změn struktury

V testovací síti u vedoucího práce byly IP adresy připojeným zařízením přidělovány DHCP serverem bez rezervace IP adres na zařízení. Při připojení nové ip adresy, která nebyla uložena v seznamu známých adres, docházelo k detekci neznámé adresy a po začátku komunikace k detekci neznámých spojení. Graf17 ukazuje počty připojení neznámých zařízení agregované po hodinách během testovacího období. Z grafu lze poznat, že před sedmou hodinou ráno a po deváté hodině večer nebyla během testovacího období neznámá zařízení nově připojována.

Ukázka zaznamenaného připojení a odpojení neznámé adresy:

```
2016-04-19 19:38:17,051 - ANOMALY - Connected
 unknown IP address: 192.168.1.218-UTC
 in time: 2015-11-19 17:33
2016-04-19 19:38:23,517 - ANOMALY - Disconnected
 unknown IP address: 192.168.1.218-UTC
 in time: 2015-11-19 19:31
```



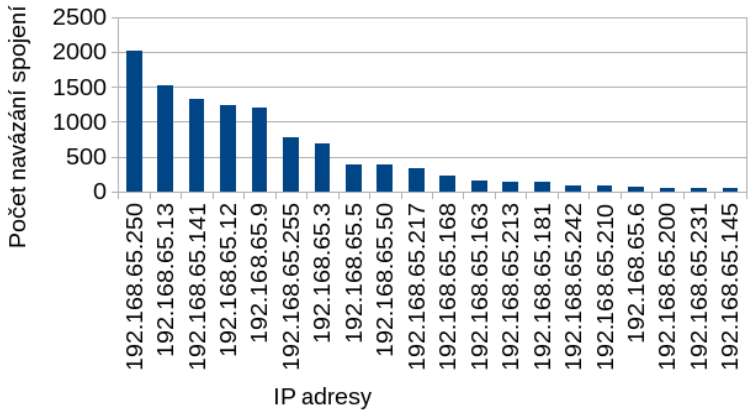
Obrázek 17: Počet připojení IP adres v testovací síti

Při vyhodnocování změn ve firemní síti byl modul v jednom měření spuštěn na rozsahu adres firemních serverů, ve druhém měření na rozsahu vyhrazeném pro připojování zařízení zaměstnanců. V serverovém rozsahu jsou adresy nastaveny staticky a připojení neznámé adresy během fáze vyhodnocování lze považovat za možný problém, který je nutné dále vyhodnotit. V síti serverů se statickým nastavením adres byly všechny adresy a spojení během učící fáze určeny za známé, proto byly exporty těchto adres a spojení přidány do souborů se známými adresami a spojeními. Vyhodnocování probíhalo od 3. 4. 2016 do 24. 4. 2016. V této době v serverové síti nedošlo k nahlášení žádného neznámého spojení.

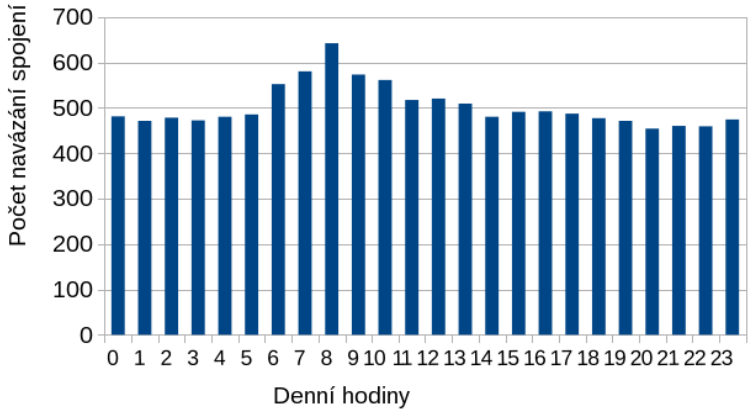
V rozsahu adres pro zařízení zaměstnanců jsou adresy zařízením přidělovány DHCP serverem bez rezervace adres na konkrétní zařízení. Připojovaná a odpojovaná zařízení během fáze učení jsou považována za neznámá a připojování a odpojování neznámých zařízení a spojení je očekáváno. Z fáze učení bylo v tomto rozsahu zjištěno 7 adres a 10 spojení mezi adresami, která byla detekována po celou dobu fáze učení a jsou považovány jako známé. Záznam z logovacího souboru o připojování neznámých zařízení během fáze vyhodnocení byl využit pro vytvoření statistiky připojování a odpojování zařízení a spojení během dne.

Za testované období 3. 4. 2016 - 24. 4. 2016 bylo v logu z vyhodnocování v rozsahu sítě pro zařízení zaměstnanců zaznamenáno celkem 12066 připojení ze 110 unikátních IP adres. Při znalosti sítě a jejího využití byla očekávána výrazně nižší hodnota. Z logu byl vyfiltrován a sestaven diagram počtu připojení během testovaného období pro jednotlivé IP adresy. 782 připojení spadalo na broadcast adresu[6], další adresy s vysokým počtem byly identifikovány z nastavení routeru pro sledovanou síť. Zjištěna byla rezervace několika IP adres z tohoto rozsahu pro zařízení ve vývoji projektů. Tato zařízení v pravidelných intervalech navazují a ukončují síťová spojení (detekováno jako připojení a odpojení) a vygenerovala tak vysoké množství připojení. Na základě tohoto vyhodnocení bylo doporučeno projektová zařízení přesunout do síťového rozsahu určeného projektům. Graf 18 zobrazuje 20 nejvíce připojovaných IP adres z nichž nejvíce používané patřily právě projektovým zařízením, IP kamerám a dvěma síťovým tiskárnám. Všechny ostatní adresy měly méně než 50 připojení během sledovaného období. Graf 19 zobrazuje počty připojení do sítě agregovaných po hodinách během sledovaného období. Patrná je vysoká základní hodnota během celého dne obsahující téměř 500 spojení na hodinu za sledovaných 21 dní. Tato hodnota je způsobena adresami identifikovaných zařízení s pravidelným navazováním spojení. V grafu lze vidět zvýšení v dopoledních hodinách způsobené příchody zaměstnanců do práce.

[6]Broadcast adresa je adresa, na které všechna zařízení v síti přijímají zprávy.



Obrázek 18: Počet připojení IP adres

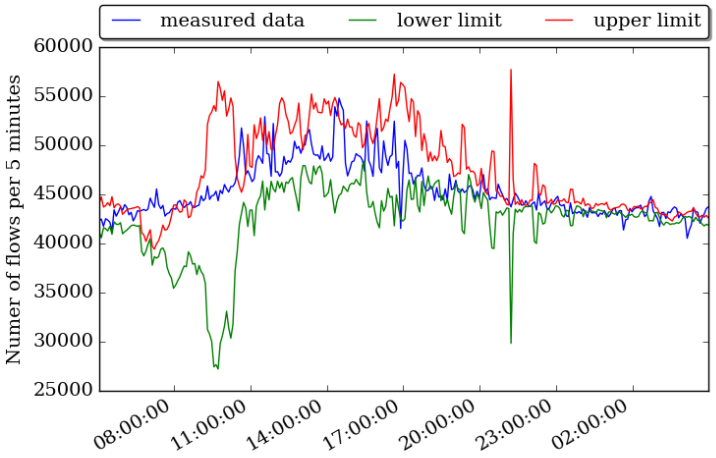


Obrázek 19: Počet připojení IP adres po hodinách

6.3 Zhodnocení testování

Při testování byly odhaleny dvě hlášené anomálie v počtu toků v testovací síti a jedna ve firemní síti. Vyhodnocování bylo účinné na zařízeních s přibližně konstantním počtem toků v provozu i na zařízeních vykazujících změny v počtu toků závislé na denní a týdenní periodě. Limitující je využití na zařízeních komunikujících v nepravidelných vysokých výkyvech v počtu toků, které jsou modulem hlášeny jako anomálie. Důležitým poznatkem z testování byla potřeba kvalitních referenčních dat bez síťových anomálií pro inicializaci výpočtu Holt-Wintersovou metodou. Při použití referenčních dat s výskytem anomálie byla při vyhodnocování snížena schopnost detekce a zvýšené množství falešně pozitivních hlášení. Na obrázku 20 je vidět výstup vyhodnocování ovlivněný anomálií ve fázi učení.

Ze záznamů o připojování a odpojování neznámých zařízení a spojení nebyla během testování ve firemní serverové síti objevena žádná anomálie. Ze záznamů o připojování a odpojování zařízení a spojení mezi zařízeními v rozsahu pro zařízení zaměstnanců byla vytvořena statistika, která ukázala na síťová zařízení, která měla být připojena v jiném síťovém rozsahu. Ve statistice počtu připojení zařízení agregované po hodinách byl ve firemní síti i v testovací síti vidět vývoj připojování zařízení v průběhu dne.



Obrázek 20: Projev výskytu anomálie ve fázi učení.

Závěr

Cílem závěrečné práce bylo vytvořit modul pro systém NEMEA, který zpracovává informace ze síťových toků. Z těchto informací generuje graf komunikace mezi jednotlivými uzly sítě, případně podsítí. Ze vzniklých struktur umožňuje sledování vývoje sítě a detekci neobvyklých jevů v síti. Součástí práce je analýza zpracování síťového provozu na základě toků. Zpracována byla také analýza statistických metod pro detekci anomálií a metod založených na klouzavém průměru, umožňujících předpověď hodnoty v časových řadách využívané při vyhodnocování anomálií.

Shrnutí průběhu a přínosů práce

Před návrhovou a implementační částí bylo nutné nastudovat strukturu a fungování systému NEMEA. Důležité bylo pochopit modulární koncept systému a komunikaci mezi moduly. Dále bylo nutné nastudovat způsoby monitorování a detekce neobvyklých jevů v síti na základě zpracování informací ze síťových toků.

V modulu byly navrženy dva způsoby detekce:

- Detekce anomálií v počtu toků na jednotlivých zařízeních, spojení mezi zařízeními a v celkovém množství toků ve sledované síti.
- Detekce připojování a odpojování zařízení a spojení mezi zařízeními s rozlišením na známá a neznámá zařízení a spojení mezi zařízeními.

Mezi vlastní přínosy pro autora práce lze jmenovat získání přehledu v problematice zpracování a vyhodnocování síťových toků, zkušeností s rozsáhlým modulárním bezpečnostním systémem a nastavením síťových prvků pro směrování firemního síťového provozu do detekčního systému. Zadání závěrečné práce bylo splněno. Byl vytvořen a otestován modul pro systém NEMEA, který zpracovává informace ze síťových toků a vytváří graf komunikace. Modul umožňuje specifikaci rozsahu sítě, nebo podsítě, která bude vyhodnoco-

vána. Detekovány jsou anomálie v množství síťových toků metodou založenou na předpovědi budoucí hodnoty množství toků Holt-Wintersovou predikční metodou. Anomálie jsou detekovány s pomocí vypočtené prahové hodnoty a chyby předpovědi. Modul dále vyhodnocuje připojování a odpojování zařízení, které přináší přehled nad změnami ve struktuře sítě a poskytuje podporu pro detekci anomálií v počtu síťových toků. Připojení neznámé adresy může znamenat připojení útočníka před začátkem útoku a detekce anomálie v počtu toků. Statistiky připojování a odpojování lze využít k dalšímu vyhodnocování chování sítě. Modul umožňuje exportovat graf struktury sítě ve formátu `gexf` pro kontrolu a případné další zpracování.

Náměty na rozšíření

Při vývoji modulu vzešly zajímavé návrhy na rozšíření funkcionality v podobě výpočtu prahové hodnoty pro detekci množství provozu „ve špičce“ a „mimo špičku“. Do modulu byla pro tento účel přidána funkce poskytující detekci provozu „ve špičce“ a „mimo špičku“ na základě předem zvolené konstanty jako prahové hodnoty. Metodu lze v budoucnu rozšířit výpočtem této hodnoty na základě vyhodnocovaných dat.

Literatura

[1] The Hierarchy of Cyber Needs [online]. Dostupné z: https://blogs.technet.microsoft.com/askpfeplat/2016/01/25/the-hierarchy-of-cyber-needs/

[2] sFlow and Netflow [online]. [cit. 2016-02-15]. Dostupné z: http://blog.sflow.com/2009/05/sflow-and-netflow.html

[3] Netflow architecture. Computer and Network Examples [online]. [cit. 2016-02-15]. Dostupné z: http://www.conceptdraw.com/How-To-Guide/netflow-architecture

[4] Ltd, O.: Exponential Moving Average (EMA)[online]. [cit. 2016-03-18]. Dostupné z: http://etfhq.com/blog/2010/11/08/exponential-moving-average/

[5] Nemea [online]. [cit. 2016-02-15]. Dostupné z: https://www.liberouter.org/technologies/nemea/

[6] Kaustubh Nyalkalkar, M. B., Sushant Sinha†; Jahanian, F.: A Comparative Study of Two Network-based Anomaly Detection Methods [online]. [cit. 2016-05-08]. Dostupné z: http://mdbailey.ece.illinois.edu/publications/infocom11_final.pdf

[7] Anukool Lakhina, M. C.; Diot, C.: Characterization of Network-Wide Anomalies in Traffic Flows[online]. [cit. 2016-04-18]. Dostupné z: http://www.cs.bu.edu/techreports/pdf/2004-020-traffic-flow-anomalies.pdf

[8] Cornell, D.: DNS Amplification Attacks [online]. [cit. 2016-04-25]. Dostupné z: https://labs.opendns.com/2014/03/17/dns-amplification-attacks/

[9] VERISIGN: UDP Flood Attacks [online]. [cit. 2016-04-28]. Dostupné z: https://www.verisign.com/en_IN/security-services/ddos-protection/denial-of-service/index.xhtml

[10] Sílvia Farraposo, P. O., Laurent Gallon: The TCP/IP and OSI Networking Models [online]. [cit. 2016-05-05]. Dostupné z: http://projects.laas.fr/METROSEC/Security_and_DoS.pdf

[11] Ferron, J.: SIP Intrusion [online]. [cit. 2016-05-08]. Dostupné z: http://www.ccctechnologies.com/2-11-15_Ask-the-Expert_SIP_Intrusion.html

[12] Farnham, G.: Detecting DNS Tunneling [online]. [cit. 2016-05-08]. Dostupné z: https://www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling-34152

[13] João M. Ceron, C. H., Klaus Steding Jessen: Anatomy of SIP Attacks [online]. [cit. 2016-05-08]. Dostupné z: https://www.usenix.org/system/files/login/articles/login1212_ceron.pdf

[14] Cynthia Bailey Lee, E. S., Chris Roedel: Detection and Characterization of Port Scan Attacks [online]. [cit. 2016-05-08]. Dostupné z: http://cseweb.ucsd.edu/~clbailey/PortScans.pdf

[15] Andreas Kind, M. P. S.; Dimitropoulos, X.: Histogram-Based Traffic Anomaly Detection. [cit. 2016-05-08].

[16] Goldberg, M. K.: Digraphs [online]. [cit. 2016-04-12]. Dostupné z: http://www.cs.rpi.edu/~goldberg/14-GT/10-directed.pdf

[17] HASHEM ALAIDAROS, A. A. M., MASSUDI MAHMUDDIN: AN OVERVIEW OF FLOW-BASED AND PACKET-BASED INTRUSION DETECTION PERFORMANCE IN HIGH SPEED NETWORKS [online]. [cit. 2016-04-06]. Dostupné z: http://www.nauss.edu.sa/acit/PDFs/f3158.pdf

[18] Institute, S.: Intrusion Detection Systems: Definition, Need and Challenges [online]. [cit. 2016-03-02]. Dostupné z: https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-systems-definition-challenges-343

[19] Institute, S.: Understanding IPS and IDS: Using IPS and IDS together for Defense in Depth [online]. [cit. 2016-04-28]. Dostupné z: https://www.sans.org/reading-room/whitepapers/detection/understanding-ips-ids-ips-ids-defense-in-depth-1381

[20] Paquet, C.: Network Security Using Cisco IOS IPS[online]. [cit. 2016-03-05]. Dostupné z: http://www.ciscopress.com/articles/article.asp?p=1336425

[21] Network Intrusion Detection Signatures [online]. [cit. 2016-02-07]. Dostupné z: http://www.symantec.com/connect/articles/network-intrusion-detection-signatures-part-one

[22] Čeleda, P.: Network Security Monitoring and Behavior Analysis [online]. [cit. 2016-04-06]. Dostupné z: http://services.geant.net/cbp/Knowledge_Base/Network_Monitoring/Documents/gn3-na3-t4-cbpd133.pdf

[23] Carter, E.: Intrusion Detection Systems [online]. 2002. Dostupné z: http://www.ciscopress.com/articles/article.asp?p=25334

[24] Odom, W.: The TCP/IP and OSI Networking Models [online]. [cit. 2016-05-05]. Dostupné z: http://www.ciscopress.com/articles/article.asp?p=1757634&seqNum=2

[25] Requirements for IP Flow Information Export (IPFIX) [online]. RFC. Dostupné z: https://tools.ietf.org/html/rfc3917

[26] RTP: A Transport Protocol for Real-Time Applications [online]. Dostupné z: https://tools.ietf.org/html/rfc3550

[27] Margaret Rouse, R. S.: Multiprotocol Label Switching (MPLS) [online]. [cit. 2016-04-06]. Dostupné z: http://searchenterprisewan.techtarget.com/definition/Multiprotocol-Label-Switching

[28] Group, N. W.: Cisco Systems NetFlow Services Export Version 9[online]. [cit. 2016-03-15]. Dostupné z: https://www.ietf.org/rfc/rfc3954.txt

[29] NetFlow vs. IPFIX[online]. [cit. 2016-02-12]. Dostupné z: https://www.whatisipfix.com/

[30] Monowar H. Bhuyan, D. K. B.; Kalita, J. K.: Network Anomaly Detection: Methods, Systems and Tools. *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS*, 2014.

[31] Rebecca Killick, I. A. E.: changepoint: An R Package for Changepoint Analysis[online]. [cit. 2016-03-18]. Dostupné z: https://www.jstatsoft.org/article/view/v058i03/v58i03.pdf

[32] Brutlag, J. D.: Aberrant Behavior Detection in Time Series for Network Monitoring [online]. [cit. 2016-02-12]. Dostupné z: http://usenix.org/legacy/publications/library/proceedings/lisa2000/full_papers/brutlag/brutlag_html/index.html

[33] Gerhard M UNZ, G. C.: APPLICATION OF FORECASTING TECHNIQUES AND CONTROL CHARTS FOR TRAFFIC ANOMALY DETECTION [online]. [cit. 2016-05-05]. Dostupné z: http://www.net.in.tum.de/fileadmin/TUM/members/muenz/documents/muenz08control-charts.pdf

[34] StatTrek: Statistics and Probability Dictionary [online]. [cit. 2016-05-08]. Dostupné z: http://stattrek.com/statistics/dictionary.aspx?definition=residual

[35] Alexander G. Tartakovsky, I. B. L. R. R. B. B., Senior Member; Kim, H.: A Novel Approach to Detection of Intrusions in Computer Networks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods [online]. [cit. 2016-05-05]. Dostupné z: https://www.researchgate.net/profile/Alexander_Tartakovsky/publication/3319811_A_novel_approach_to_detection_of_intrusions_in_computer_networks_via_adaptive_sequential_and_batch-sequential_change-point_detection_methods/links/0deec535e6bc973ef1000000.pdf

[36] Anna Sperotto, R. S. C. M. A. P., Gregor Schaffrath; Stiller, B.: An Overview of IP Flow-Based Intrusion Detection. *IEEE COMMUNICATIONS SURVEYS and TUTORIALS*, 2010.

[37] Ratnadip Adhikari, R. K. A.: An Introductory Study on Time Series Modeling and Forecasting [online]. [cit. 2016-04-18]. Dostupné z: http://www.realtechsupport.org/UB/SR/time/Agrawal_TimeSeriesAnalysis.pdf

[38] Josef Arlt, E. R., Markéta Arltová: ANALÝZA EKONOMICKÝCH ČASOVÝCH ŘAD S PŘÍKLADY [online]. [cit. 2016-02-13]. Dostupné z: http://nb.vse.cz/~arltova/vyuka/crsbir02.pdf

[39] Moving Averages - Simple and Exponential [online]. [cit. 2016-02-07]. Dostupné z: http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_averages

[40] Taylor, J. W.; McSharry, P. E.: Short-Term Load Forecasting Methods: An Evaluation Based on European Data. *IEEE TRANSACTIONS ON POWER SYSTEMS*, 2007.

[41] Hagberg, A. A.; Schult, D. A.; Swart, P. J.: Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Srpen 2008, s. 11–15.

[42] Peixoto, T. P.: The graph-tool python library. *figshare*, 2014, doi:10.6084/m9.figshare.1164194. Dostupné z: http://figshare.com/articles/graph_tool/1164194

[43] Foundation., P. S.: High-performance container datatypes [online]. [cit. 2016-04-07]. Dostupné z: https://docs.python.org/2/library/collections.html#collections.deque

[44] developer team, N.: NetworkX [online]. [cit. 2016-04-07]. Dostupné z: https://networkx.github.io/index.html

[45] Hunter, J. D.: Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, ročník 9, č. 3, 2007: s. 90–95.

[46] community, T. S.: scipy.optimize.fmin_l_bfgs_b [online]. [cit. 2016-03-05]. Dostupné z: http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.optimize.fmin_l_bfgs_b.html

[47] Jones, E.; Oliphant, T.; Peterson, P.; aj.: SciPy: Open source scientific tools for Python. 2001–, [Online; accessed 2016-05-08]. Dostupné z: http://www.scipy.org/

[48] Bastian, M.; Heymann, S.; Jacomy, M.: Gephi: An Open Source Software for Exploring and Manipulating Networks. 2009. Dostupné z: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

[49] Foundation, P. S.: Python Language Reference. Version 2.7. Available at http://www.python.org.

[50] Amy Ward, K. R., Peter Glynn: Internet service performance failure detection. *Performance Evaluation Review*, 1998.

[51] Ward, G.: Installing Python Modules [online]. [cit. 2016-05-08]. Dostupné z: https://docs.python.org/2/install/

Příloha A

Seznam použitých zkratek

GHRG Generalized hierarchical random graph

NEMEA Network measurements analysis

IDS Intrusion detection system

HIDS Host based Intrusion detection system

NIDS Network based Intrusion detection system

IPS Intrusion protection system

HIPS Host based Intrusion protection system

NIPS Network based Intrusion protection system

IPFIX IP Flow Information Export

VoIP Voice over IP

TRAP Traffic analysis platform

UniRec Unified record

IP Internet protocol

SNMP Simple network management protocol

ICMP Internet control message protocol

RPC Remote procedure call

OS Operační systém

VPN Virtual private network

PCAP Packet capture

DNS Domain name server

SIP Session initiation protocol

SSH Secure shell

UDP User datagram protocol

FTP File transfer protocol

HTTP Hypertext transfer protocol

DOS Denial of service

Příloha B

Instalační příručka

B.1 Požadavky na hardware a software

Modul je navržen pro provoz v operačním systému Linux. K provozu je zapotřebí nainstalovaný systém NEMEA s modulem poskytujícím rozhraní s výstupem záznamů o síťových tocích. Modul pro svůj běh vyžaduje nainstalovaný Python verze 2.7. Potřebné jsou nainstalované tyto knihovny pro Python:

- networkx
- matplotlib
- scipy
- time
- datetime
- ipaddress
- json
- logging
- collections

Pro zobrazení grafických výstupů je potřeba prohlížeč souborů `png` grafického formátu. Pro zobrazení výstupu struktury grafu je potřeba prohlížeč souborů s podporou `gexf` formátu. V OS Linux lze využít například program `Gephi`.

Minimální požadavky na hardware jsou závislé na způsobu použití modulu a množství zpracovávaných dat. Doporučené je CPU: 2GHz, RAM: 2GB, volný prostor na disku: 1GB.

B.2 Instalace

Instalace modulu probíhá v následujících krocích:

1. Instalace nástrojů a závislostí uvedených v sekci B.1.

2. Modul je instalován spuštěním `python setup.py install` z adresáře s modulem, tím je modul instalován do výchozí cesty pro instalaci podle nastavení v OS. Instalace je prováděna pomocí nástroje `distutils`, další možností nastavení instalace poskytované nástrojem `distutils` lze nalézt v [51]

Příloha C

Uživatelská příručka

Základní informace o funkci a možnosti spuštění modulu jsou uvedeny v nápovědě, která je vyvolána spuštěním modulu s parametrem `-h` nebo `--help`. V této kapitole je rozšíření těchto informací.

C.1 Spuštění

Pro spuštění modul vyžaduje nastavení parametrů pro poskytnutí přístupových práva pro vytváření, čtení a zápis souborů a adresářů. K počátečnímu nastavení lze využít následující parametry:

- - -learning: Nepřijímá vstupní hodnoty. Spouští modul v režimu učení. Modul v režimu učení sbírá a zpracovává data ze dvou týdnů provozu pro pozdější využití v režimu vyhodnocování. Po dvou týdnech jsou data uložena do souboru a modul je přepnut do režimu vyhodnocování.
- - -ip-range: Rozsah IP adres ve formátu IP1-IP2 pro podrobné zpracování. Ostatní IP adresy jsou sdruženy a vyhodnocovány jako IP 0.0.0.0.
- - -no-structure-detect: Nepřijímá vstupní hodnoty. Parametr pro vypnutí detekce připojování a odpojování zařízení. Modul vyhodnocuje a hlásí anomálie pouze pro počty toků.
- - -traffic-state-threshold: Přijímá hodnotu typu float. Výchozí hodnota 0. Určuje počet toků na jedno pěti minutové okno časové řady, jako hranici pro přepnutí mezi režimem provozu „ve špičce“ a „mimo špičku“.
- - -detections-trigger: Přijímá hodnotu typu int. Výchozí hodnota 1. Určuje počet oken s detekovanou změnou v provozu před nahlášením anomálie.
- - -scaling-factor: Přijímá hodnotu typu float. Výchozí hodnota 3. Parametr pro přizpůsobení prahových hodnot pro vyhodnocení anomálie. Doporučené je nastavení hodnoty mezi 2 - 3.

- - -logger-severity: Přijímá řetězec, očekávány jsou hodnoty info nebo anomaly. Výchozí hodnota anomaly. Slouží k nastavení úrovně pro logování.
- - -whitelist-path: Přijímá řetězec. Výchozí hodnota whitelist.txt. Nastavuje cestu pro soubory s whitelisty na IP adresy a spojení mezi IP adresami. Modul na dané cestě podle zvoleného názvu vytvoří soubory pro whitelisty s prefixy IP- a Connection-. Pro whitelist.txt to bude IP-whitelist.txt, Connection-whitelist.txt. Dále na zadané cestě modul vytvoří soubory Connection-ALL.txt a IP-ALL.txt s uloženými všemi adresami a spojeními z fáze učení.
- - -plot-interval: Přijímá hodnotu typu int. Výchozí hodnota 1. Nastavuje počet dní sloužících jako interval pro export kontrolních grafů.
- - -file-path: Nastavuje cestu k souboru s referenčními daty. Výchozí cesta data/learned.json. Pokud je modul spuštěn v režimu učení, určuje tato cesta místo uložení naučených dat. Pokud je modul spuštěn v režimu vyhodnocování, určuje cestu k importu naučených dat.
- - -output-path: Přijímá řetězec. Výchozí hodnota output. Nastavuje cestu pro výstup vyhodnocování.
- - -quiet: Neočekává vstup. Potlačuje textový výstup do terminálu.

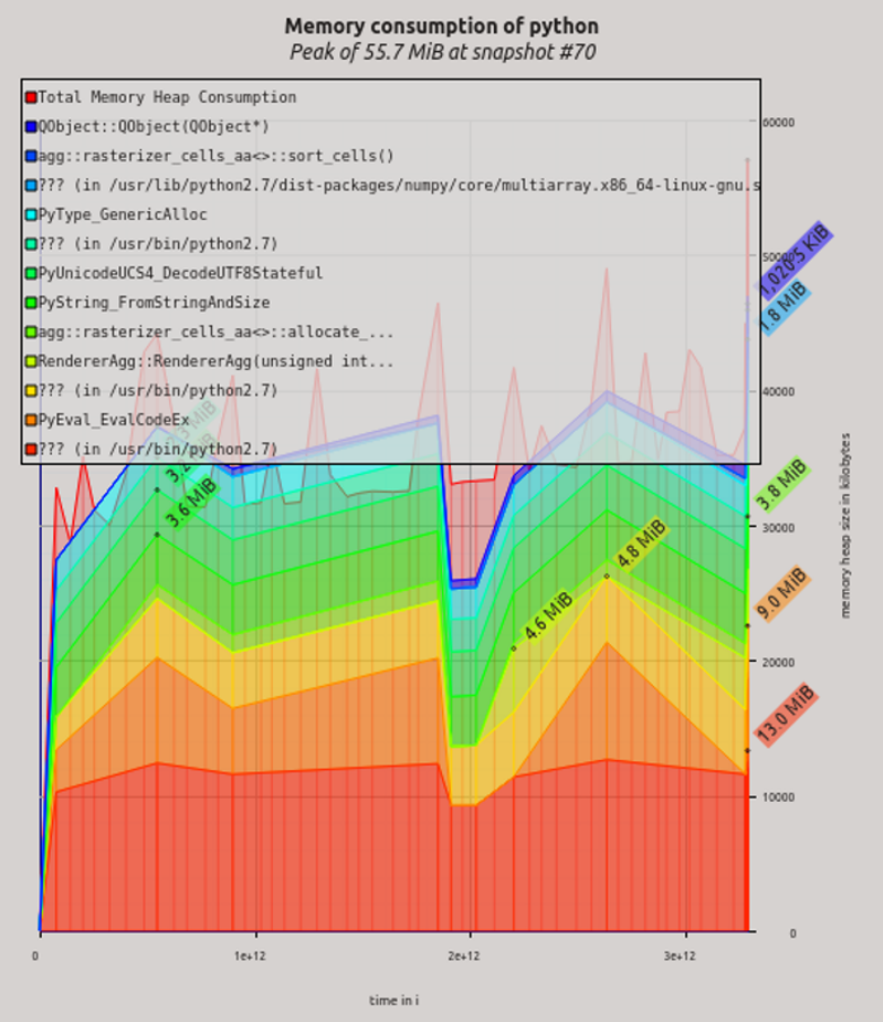
Komunikační rozhraní v modulu poskytuje importovná knihovna libtrap. Pro spuštení modulu je povinný pouze parametr `-i` libtrap knihovny pro specifikaci rozhraní. Nápovědu k libtrap knihovně lze zobrazit při spuštění modulu s parametrem `-h 1`.

Níže je příklad spuštění modulu v režimu vyhodnocování, kde jsou změněny následující hodnoty oproti výchozím. Načtení naučených dat z data/data.csv, ukládání výstupů do adresáře output, uložení whitelist souborů do adresáře data, kde budou pojmenovány jako IP-whitelist.txt a Connection-whitelist.txt. Soubory všech adres a spojení z fáze učení budou uloženy do adresáře data pod názvy Connection-ALL.txt a IP-ALL.txt.

```
graphflow.py --interface u:datasource
--file-path data/data.csv
--output-path output
--whitelist-path data/whitelist.txt
```

Příloha D

Profilování využití paměti



Obrázek 21: Test využití heap.

Příloha **E**

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src` ..... adresář se se zdrojovými soubory modulu
  - `log_filters.txt` příkazy použité pro filtrování informací z logovacího souboru
  - `impl` ..... zdrojové kódy implementace
    - `setup.py` ..... instalační soubor
    - `graphflow.py` ..... hlavní část implementace modulu
    - `hwt.py` ..... implementace Holt-Wintersovy metody
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
  - `DP_Vicher_Tomáš_2016.tex` ..... zdrojová forma práce
- `text` ..... text práce
  - `DP_Vicher_Tomáš_2016.pdf` ..... text práce ve formátu PDF
  - `DP_Vicher_Tomáš_2016.ps` ..... text práce ve formátu PS