VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE SÍŤOVÝCH ÚTOKŮ POMOCÍ STATISTICKÝCH MODELŮ NAD NETFLOW DATY

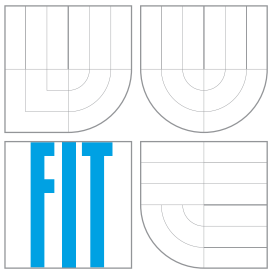
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAKUB ČEGAN

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE SÍŤOVÝCH ÚTOKŮ POMOCÍ STATISTICKÝCH MODELŮ NAD NETFLOW DATY

NETWORK ATTACKS DETECTION USING STATISTICAL MODELS WITH NETFLOW DATA

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. JAKUB ČEGAN
AUTHOR

VEDOUCÍ PRÁCE — Ing. JIŘÍ NOVOTŇÁK
SUPERVISOR

BRNO 2012

Abstrakt

Diplomová práce popisuje vybrané metody detekce síťových útoků pomocí aplikace statistických modelů nad NetFlow daty. V úvodní části popisuje některé hrozby, které často postihují počítačové sítě a jsou dobře detekovatelné v NetFlow datech. Práce zároveň prezentuje samotnou technologii NetFlow včetně protokolu a architektury. V teoretické části jsou dále podrobně popsány statistické metody použitelné pro detekci útoků s důrazem na metodu ASTUTE. Další část se věnuje představení nástrojů použitých k implementaci metod pomocí pluginů programu NfSen. Následuje podrobný popis implementace pluginů a jejich následného testování včetně provedených simulovaných útoků.

Abstract

This diploma thesis describes several selected network attacks detection method using statistical models with NetFlow data. First are described several well known and threats for computer networks, which are easily detectable in the NetFlow data. Thesis also introduce and present the NetFlow technology including its protocol and architecture. The theoretical part of the thesis describes statistical methods with focus on the ASTUTE method, that can be used for an anomaly detection. Following part introduces tools used for method implementation as the NfSen plugins. Last parts of the thesis describe in detail implementation of the plugins and following plugins testing which included simulated network attacks.

Klíčová slova

ASTUTE, detekce, ochrana počítačové sítě, NetFlow, NfSen, plugin, statistický model

Keywords

ASTUTE, detection, network protection, NetFlow, NfSen, plugin, statistical model

Citace

Jakub Čegan: Detekce síťových útoků pomocí statistických modelů nad netflow daty, diplomová práce, Brno, FIT VUT v Brně, 2012

Detekce síťových útoků pomocí statistických modelů nad netflow daty

Prohlášení

Prohlašuji, že jsem tuto předkládanou diplomovou práci vypracoval samostatně pod vedením pana Ing. Jiřího Novotňáka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jakub Čegan
23. května 2012

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Jiřímu Novotňákovi za čas věnovaný konzultacím k této práci a veškerou další pomoc. Dále bych děkuji Ing. Jiřímu Tobolovi za poskytnutí možnosti testovat pluginy v jeho síti a také všem ostatním za jimi poskytnuté odborné rady a podnětnou diskuzi.

Obsah

Seznam obrázků

Kapitola 1

Úvod

V dnešní době je možné pozorovat mnohem větší rozmach počítačových sítí než v letech předchozích. Dramaticky se zvyšuje jejich počet, rychlost a velikost. Součástí sítí se stávají masově nová zařízení jako jsou chytré telefony. Dalším trendem dnešní doby je tvorba rozsáhlých webových služeb obsahujících více uživatelů a informací, než bylo kdykoliv v minulosti představitelné. Tvorba tohoto světa počítačových sítí je permanentně ovlivňována stále novými omezeními a výzvami.

Jednou ze zásadních výzev je zaručení provozuschopnosti tak rozsáhlé infrastruktury a zajištění její bezpečnosti a bezpečnosti jejich uživatelů před nástrahami a hrozbami počítačové sítě. Dnes již není při tak masivním provozu možné úspěšně kontrolovat jednotlivé pakety všech zařízení v síti. Proto začala být využívána technologie NetFlow, která umožňuje snadno kontrolovat vysokorychlostní a rozsáhlé sítě. Také je potřeba zmínit, že vzhledem k tomu, že tato technologie abstrahuje od práce s pakety, zachovává dnes tolik důležité soukromí uživatelů. NetFlow dává základ pro tvorbu detekčních a kontrolních mechanizmů založených na pravidlech a nebo založených statistických metodách pracujících s toky v počítačové síti.

A právě statistické metody jsou obsahem této práce. Jejich podstatnou výhodou je jejich matematický základ, takže jsou schopny popsat obecně platné jevy a odhalovat tak dosud neznámé anomálie a nově vytvořené útoky, které ještě nebyly analyzovány a nebyla pro ně vytvořena žádná detekční pravidla. Rovněž jsou tyto metody imunní před modifikacemi útoků, které útočníci často činí, aby je skryli před nástroji chránícími počítačové sítě.

Předmětem této práce je vypracování pohledu na statistickou analýzu síťových toků. Jsou zde rozebrány některé metody pro statistickou analýzu, které by mohly pomoci s odhalováním anomálií a útoků v počítačových sítích. Takovou metodou je například ASTUTE, která vykazuje dobré výsledky zejména na neznámých útocích [24]. Každá ze statistických metod byla zpracována jako plugin pro aplikaci NfSen, který demonstruje její funkčnost a úspěšnost, případně neúspěšnost detekce jednotlivých útoků.

V první kapitole této práce jsou stručně popsány nejčastější hrozby objevující se v dnešních počítačových sítích. Kapitola druhá obsahuje popis monitorování sítí pomocí síťových toků a technologie NetFlow. Stručně je popsán vlastní protokol NetFlow ve verzi 9 a také je přiblížena architektura celého systému pro práci se síťovými toky. Třetí kapitola popisuje statistické metody, které mohou být použitelné ke sledování provozu v počítačových sítích a zaměřuje se na metodu ASTUTE. Kapitola čtvrtá se zaměřuje na vlastní architekturu systému NfSen, do kterého jsou tyto statistické metody ve formě pluginů zapracovány. Také popisuje veškeré nástroje používané systémem NfSen pro svůj běh. Následuje stručné shrnutí práce a popis jejího dalšího využití.

Kapitola 2

Hrozby v počítačových sítích

Kapitola popisuje některé z aktuálních hrozeb vyskytujících se v počítačových sítích. Je provedena jejich kategorizace v rámci možností a rozsahu práce. U vybraných z nich je pak detailněji přiblížen jejich průběh. Také je u každé hrozby popsán cíl, kterého se snaží v případě svého úspěchu dosáhnout.

2.1 Skenování portů

Skenování portů je jedním z nejrozšířenějších typů útoků proti strojům v počítačových sítích a představuje podstatnou část provozu v síti internet [15]. I přesto, že se může občas jednat o poměrně neškodnou zábavou nezkušených uživatelů, není vhodné jej přesto brát na lehkou váhu, protože v rukou zkušeného útočníka je prvním krokem k určení operačního systému stroje a dalších užitečných informací, které využije při dalším útoku. Výsledkem může být až ovládnutí systému a jeho zdrojů, které následně může využít pro další útoky. Zkušení útočníci se také snaží zastřít svou činnost, například rozprostřením skenování do delšího časového okna, případně použijí větší množství skenujících strojů [15]. Skenování portů lze rozdělit do tří hlavních skupin dle provedení skenu a svého cíle [15], které jsou stručně charakterizovány v následujících odstavcích.

2.1.1 Vertikální skenování

Vertikální sken představuje skenování blíže nespecifikovaného množství portů jednoho konkrétního stroje v síti. V případě naivního provedení je to nejsnadněji detekovatelný druh skenu, protože jsou aplikovány pouze lokální detekční mechanizmy [15].

2.1.2 Horizontální skenování

V případě horizontálního skenování se jedná o testování jednoho portu, například portu číslo 22, kde se běžně nachází SSH, přes mnoho strojů, typicky v jedné podsíti. Velmi často dochází k masivnímu skenování určitého portu po veřejném oznámení nalezení nové zranitelnosti služby, která se na tomto portu standardně vyskytuje [15].

2.1.3 Blokové skenování

Takzvaný blokový sken je kombinací obou dříve zmíněných přístupů a útočníci jej často používají pro nalezení strojů v síti, které obsahují některou ze seznamu jim dobře známých zranitelností, pro něž mají připraveny další útoky [15].

2.1.4 Charakteristický zástupce skenování

I přesto, že existuje nepřeberné množství programů určených pro skenování portu, je možné bez zaváhání vybrat jejich charakteristického zástupce [15]. Jedná se o program Nmap autora vystupujícího pod přezdívkou Fyodor. Velkou výhodou tohoto programu je kromě jeho širokých možností skenování také webová stránka [17] s návody a dokumentací.

Pokud by měl být uveden charakteristický zástupce skenování portů, pak by to byl dle počtu použití jednoznačně TCP SYN sken [17], který je dostatečně jednoduchý, rychlý a efektivní, aby mohl být masově používán. Umožňuje také jasně rozlišit mezi otevřeným, uzavřeným a filtrovaným portem. Je zde využito principu sestavování TCP spojení. Na skenovaný port je odeslán paket s nastaveným příznakem SYN. Uzavřený port zpět odpoví paketem s příznakem RST. Port otevřený odpovídá SYN/ACK a v málo se vyskytujících případech, způsobených takzvaným simultánním otevřením spojení [17], SYN. Port je označen jako filtrovaný, pokud nedorazí ani jedna z odpovědí po několikrát opakovaném odeslání SYN paketu. Také je označen jako filtrovaný, dorazí–li některý z datagramů ICMP protokolu označujícího nedosažitelnost [17].

2.2 Botnet

Botnet představuje množinu softwarových robotů tvořenou z infikovaných strojů a spojenou do jedné sítě za účelem využití jejich výpočetního času a konektivity bez vědomí majitele k převážně ilegální činnosti, jako je rozesílání spamu a útoky na servery. Na černém trhu je také možné koupit od tvůrců výpočetní čas v jejich botnetu pro vlastní účely [19].

2.2.1 Anatomie a životní cyklus botnetu

Celý botnet se skládá ze dvou součástí. První jsou roboti, někdy také označovaní jako zombie a řídící středisko botnetu (označováno C&C) pomocí kterého majitelé botnetu rozesílají požadavky a aktualizace jednotlivým strojům. Toto se děje pomocí protokolu HTTP(S) [10], nebo IRC [27]. Komunikace bývá zašifrována, aby se zabránilo převzetí botnetu konkurenční skupinou, nebo jeho zničení [10] některou z organizací zabývajících se jejich výzkumem. Některé z botnetů využívají distribuovanou architekturu (P2P) nebo dokonce některou z veřejných P2P sítí [10], aby tak snížili možnost odstavení sítě zničením kontrolního střediska. Životní cyklus botnetu je pak možné shrnout do několika stručných bodů:

1. Tvůrce botnetu vypouští do světa malware a infikuje počítače běžných uživatelů a vytváří tak první boty svého nového botnetu.

2. Tito se při nejbližší příležitosti přihlásí k řídícímu středisku botnetu a obdrží své první úkoly, mezi něž patří nalezení dalších potencionálních obětí v síti, do které naleží.

3. V tento okamžik již může tvůrce botnetu využívat jeho zdroje ke svým záměrům, jako je rozesílání spamu a prodej procesorového času třetím stranám.

4. Botnet je nadále spravován a vylepšován svým tvůrcem. Může teoreticky existovat po libovolně dlouhou dobu, případně do okamžiku, kdy jsou jeho boti převzati konkurencí, nebo je celý zničen výzkumníky.

2.2.2 Botnet Chuck Norris

Botnety však nevyhledávají své nové stroje jen mezi PC a servery. Existují i specifické typy botnetů, které napadají routery a další zařízení pomocí chyb v zastaralém firmware založeném na Linuxu [27]. Mezi ně je možné zařadit i nedávno objevený botnet Chuck Norris. Jeho největším nebezpečím je, že napadá vlastní infrastrukturu počítačové sítě mimo dosah antivirových programů [27]. K jeho odhalení napomohla charakteristika komunikace s řídícím centrem botnetu a stahování aktualizací botnetu je jedním z klíčových způsobů jejich odhalení [27].

2.3 Útok odepřením služby

Útok odepřením služby je, jak jeho název napovídá, veden za účelem znepřístupnění stroje a služby na něm běžící, typicky jde o DNS, webový nebo emailový server, na různě dlouhou dobu pro její legitimní uživatele. Útok bývá často proveden za účelem nátlakové akce na provozovatele, případně majitele napadené služby. Dalším cílem, kterého je možné dosáhnout skrze tento útok, je vynucení restartu napadeného stroje poté, co na něj byl útočníkem nahrán škodlivý kód.

Americký US-CERT tým, jehož úkolem je zlepšování počítačové bezpečnosti v USA, popisuje ve své starší, ale stále platné definici útok odepřením služby pomocí následujících čtyřech symptomů [18]:

- zpomalení sítě při běžných činnostech (otevírání souborů, přístup na webové stránky)
- nedostupnost některých webových stránek
- nemožnost přístupu na jakoukoliv webovou stránku
- dramatický nárůst množství obdrženého spamu

Útoky tohoto typu jsou nejčastěji děleny dle dvou základních kritérií do následujících skupin a tak je tomu i v této práci. První kritériem je způsob provedení samotného útoku na stroj. Druhým pak je počet účastnících se útočníků. V závěru je pak uveden charakteristický představitel tohoto typu útoku.

2.3.1 Klasifikace dle způsobu provedení

Prvním ze způsobů provedení útoku je klasický útok hrubou silou, většinou nazývaný *Flood*. Smyslem tohoto útoku je doslova zaplavení napadeného stroje velkým množství požadavků [11] tak, že je znemožněna jeho funkce pro legitimní uživatele systému. Tyto útoky buď pouze blokují stroj po dobu svého trvání, nebo mohou způsobit vyčerpání prostředků a jeho pád. Typickými představiteli pak jsou SYN flood [11], UDP flood a HTTP flood [25].

Druhou a mnohem sofistikovanější metodou je zneužití chyby v implementaci služby běžící na napadeném stroji [11]. Útok je někdy označován jako *Nuke*. Ke shození počítače nedochází důsledkem vyčerpání systémových zdrojů, ale zasláním jednoho vhodně upraveného paketu nebo dotazu, který způsobí vyvolání neošetřeného stavu a následný pád stroje. Dobrou ilustrací tohoto typu útoku je WinNuke, který napadal stroje s operačním systémem MS Windows 95 a NT [31].

2.3.2 Klasifikace dle počtu útočníků

V případě klasifikace dle počtu útočníků je prvním zástupcem klasický útok *DoS*. Tento typ útoku je dnes možné použít pouze v případě útoku vyžívajícího zranitelnost služby, protože jeden útočník nemůže v naprosté většině případů získat sám tak velkou kapacitu linky, aby dokázal zahltit stroje v datacentrech.

Dalším a dnes hojně využívaným je distribuovaný útok *DDoS*. Do útoku mohou být zapojeny počítače infikované malwarem a sdružené do sítě nazývané botnet. Nebo se mohou spojit i sami uživatelé a po stažení nástroje *LOIC*, se stát dobrovolně součástí útoku [21].

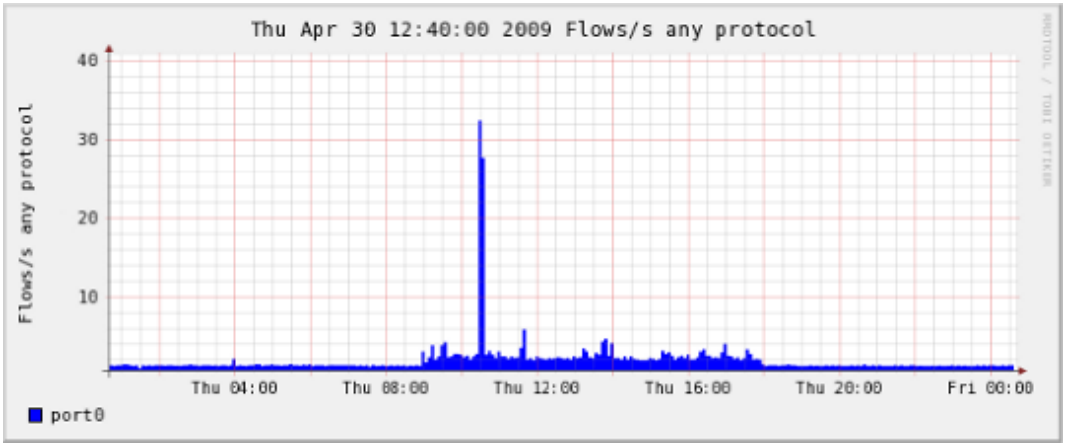
Síla útoku logicky roste s počtem zapojených strojů a jejich konektivitou. Mohou proto blokovat stroje napadeného subjektu v řádech hodin i dnů, případně donutit tak napadený subjekt své stroje odstavit, dokud není útok ukončen. Dnes se začínají objevovat pokusy bojovat s DDoS útoky pomocí cloudových řešení, která v případě útoku poskytnou větší kapacitu pro zvládnutí zvýšeného provozu [38].

Poslední ze skupiny těchto útoků je takzvaný distribuovaný reflektovaný útok *DRDoS*. Útočník odešle na stroje třetích stran množství požadavků, kde na místo zdrojové adresy uvede adresu cíle. Následně stroje třetí strany zahltí svými odpověďmi napadený stroj. Mezi největší výhody tohoto typu útoku patří zesílení vlastní síly a zakrytí útočících strojů.

Jedním z principů útoku je zneužití TCP handshake [8]. Útočník zasílá velké množství paketů s nastaveným příznakem SYN a jako zdrojovou IP adresou nastaví adresu cíle útoku na stroje sloužící jako reflektor. Na každý jeden paket odeslaný na tento reflektor jsou z nich vygenerovány proti oběti čtyři SYN/ACK pakety [8].

2.3.3 Charakteristický zástupce HTTP flood

Jedním z charakteristických představitelů je HTTP flood útok. Používán je proti serverům s cílem znepřístupnit na nich běžící webové stránky. Představuje jeden z nejjednodušších a přesto nejničivějších útoků. Principem tohoto útoku je zasílání běžných či nesmyslných požadavků. Jeho nebezpečnost spočívá v obtížné odlišitelnosti od reálného provozu, protože tyto požadavky jsou z hlediska firewallu naprosto legitimní a není tedy důvod k jejich filtraci. Server poté nemá dostatek zdrojů k obsloužení legitimních uživatelů.



Obrázek 2.1: Ukázka HTTP flood útoku v NetFlow [26].

Pro útok je možné použít jednoduchou metodu zaplavení portů 80 a 443 velkým množstvím požadavků [26]. Výsledek takového útoku je zachycen na obrázku 2.1. Nebo lze pou-

žít sofistikovaný přístup, který nabízí například program Slowloris. Tento program otevírá a udržuje co nejdéle velké množství spojení na cílovém web serveru pomocí odesílání nekompletních HTTP požadavků [23].

2.4 Útoky na SSH

Protože je SSH aktuálně nejpoužívanějším službou a protokolem pro bezpečný vzdálený přístup k počítačovým systému, tak je také nejvíce ohroženou službou v počítačové síti. Pokud nebude uvažována zastaralá první verze protokolu obsahující zásadní neopravitelnou bezpečnostní chybu [6], pak je běžně používána proprietární implementace SSH2 a varianta s otevřeným zdrojovým kódem OpenSSH.

Nejrozšířenější metodou útoku jsou různé modifikace hádání přístupového hesla a jména, které se liší v podstatě jen sofistikovaností přístupu. Je možné se setkat s naivními útoky, kdy jsou hrubou silou generována hesla z jediného stroje. Tato metoda má pochybnou úspěšnost vzhledem k výpočetní náročnosti, omezené kapacitě linek a časovému intervalu potřebnému k provedení jedné akce.

Vylepšením této metody je využití předem vytvořeného slovníku hesel a jejich permutací za použití substituce písmen za symboly a čísla (populární Leet [37]). Jednou z výhod specifického českého prostředí je v tomto případě potřeba specializovaných slovníků pro český jazyk. Tyto případy jsou snadno odhalitelné i když se snaží maskovat svou aktivitu, například rozmělněním celého útoku v čase [28].

Mnohem zákeřnější útoky rozkládají svou činnost nejen v čase, ale také využívají pro maskování armády počítačů, které typicky patří do botnetu. Při vhodném nastavení všech parametrů je poměrně obtížné je běžnými metodami odhalit.

Jednou z představených metod je detekce dle charakteristiky provozu v NetFlow. Takovýto útok se projevuje jako velký počet podobných toků v určitém časovém okamžiku. Jednotlivé toky od útočníka mají vždy více než 10 paketů, protože v opačném případě nemůže jít o korektní přihlášení [28]. Snížit počet falešných poplachů lze například přes rozhodovací stromy pro detekci slovníkových útoků na SSH [28]. Hledání je také možné omezit na port 22, který je výchozím pro SSH.

2.5 Malware

Jako malware je označována celá rodina škodlivých softwarů zahrnující viry, červy a trojské koně. Dnes jsou šířeny především za účelem získání citlivých dat uživatelů jako jsou přístupová hesla ke službám a elektronickému bankovnictví z napadených strojů. Druhým důvodem, který je často doplněn důvodem prvním, je snaha získat napadený stroj pro některý z mnoha botnetů.

Boj s malware je poměrně obtížný. Sestává se z blokování nebezpečných příloh emailů, blokování nebezpečných webových stránek, udržování aktualizovaných antivirových programů a v neposlední řadě z osvěty uživatelů. V případě, že dojde k rozšíření malware v počítačové sítí, je možné pozorovat skenování portů ve vnitřní síti ukazující snahu infikovaného stroje o nalezení dalších potencionálních obětí.

Kapitola 3

Monitorování provozu na základě síťových toků

Kapitola blíže popisuje pojem síťový tok, jak je definován v různých infomračních pramenech zabývajících se touto problematikou. Dále popisuje jak protokol NetFlow v9, tak se také zaobírá architekturou, na které je daný protokol používán.

3.1 Síťový tok

Pojem síťový tok je definován v normách rozličným způsobem. Například RFC 2722 jej definuje jako umělý logický ekvivalent k volání nebo spojení. Je to úsek provozu ohraničený počátečním a koncovým časem [3]. Norma pro Internet Protocol Information Export (RFC3917, dále IPFIX) jej popisuje jako množinu IP paketů procházejících pozorovaným bodem v počítačové síti po určitý časový interval a všechny pakety patřící do jednoho flow vykazují množinu shodných vlastností [22] a konečně RFC 3697, které popisuje záležitosti okolo toků a IPv6 definuje tok jako sekvenci paketů poslanou z konkrétního zdroje do určité unicastového, anycastového, nebo multicastového cíle. Tok by se měl skládat ze všech paketů v určitém transportním spojení nebo streamovaném médiu. Nicméně nemusí být mapován 1:1 na transportní spojení [22].

Pro tuto práci bude použita definice toku, která jednoznačně určuje, že síťový tok může být unikátně identifikován v určitém časovém období jako jednosměrná posloupnost paketů se shodnou zdrojovou a cílovou IP adresou, zdrojovým portem, cílovým portem a protokolem (TCP,UDP, ICMP) [28].

3.2 NetFlow

Protokol NetFlow byl vyvinut a nasazen společností Cisco Systems jako otevřený protokol pro monitorování provozu v počítačových sítích pomocí síťových toků, který poskytuje podrobný pohled do síťového provoz. Protokol ve svých počátcích fungoval jako doplňková služba pro směrovače vyráběné touto společností.

Mezi zásadní výhody použití NetFlow při správě a odhalování útoků v počítačových sítích patří schopnost pracovat na sítích s vysokou rychlostí a silným provozem [30]. Dále je důležité, že NetFlow ze své podstaty neprozrazuje žádné citlivé, či osobní údaje, které jsou součástí paketů v monitorovaných tocích. Je tedy ve shodě se zpřísňující se legislativou o ochraně osobních údajů v celé Evropské unii.

3.3 Protokol

První masově použitou verzí NetFlow protokolu byla jeho pátá verze. Aktuálně používaná devátá verze má na rozdíl od verzí předcházejících jedno velmi podstatné vylepšení. Celá struktura tohoto protokolu je tvořena pomocí šablon [33]. Hlavní výhodou výše uvedeného přístupu je, že umožňuje rozšíření NetFlow služeb bez změny původního záznamu [4]. Od této verze je také možné zaznamenávat IP adresy ve formátu IPv6. NetFlow protokol je tedy připraven na případné vyčerpání IP adres verze čtyři. Na základech protokolu verze 9 byl postaven nový protokol IPFIX, který byl prohlášen standardem Internet Engineering Task Force (IETF) [22]. Tento protokole je pokusem o sjednocení roztříštěných metod a protokolů pro práci se síťovými toky u různých výrobců hardware pro tuto technologii.

Hlavička paketu	Template FlowSet	Data FlowSet	Data FlowSet	...	Template FlowSet	Data FlowSet

Tabulka 3.1: NetFlow paket verze 9 [33].

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Verze																Počet															
Uptime systému																															
UNIX timestamp																															
Číslo sekvence																															
ID zdroje																															

Tabulka 3.2: Formát hlavičky NetFlow paketu verze 9 [33].

Pole	**Popis hodnoty**
Verze	Verze NetFlow záznamu v paketu. Verzi 9 odpovídá 0x0009.
Počet	Počet FlowSet záznamů (template i data) obsažených v paketu.
Uptime systému	Doba v milisekundách od okamžiku, kdy bylo zařízení spuštěno.
UNIX timestamp	Počet sekund od počátku Epochy (1.1.1970 00:00 UTC).
Číslo sekvence	Vzestupný čítač počítající všechny pakety exportované dotyčným zařízením. Jedná se o kumulativní hodnotu, která může být použita k identifikaci chybějícího paketu.
ID zdroje	ID zdroje je 32 bitová hodnota zaručující unikátnost pro všechny toky exportované z určitého zařízení. Formát tohoto pole se liší výrobce od výrobce. V případě produktů Cisco jsou první dva byty vyhrazeny pro budoucí rozšíření a obsahují v současné době nulové hodnoty. Třetí a čtvrtý byte zajišťují unikátnost. Kolektor potom používá IP adresu a ID pro spojení příchozího NetFlow paketu se unikátní instancí NetFlow na patřičném zařízení.

Tabulka 3.3: Popis polí v hlavičce NetFlow paketu verze 9 [33].

3.4 Architektura

NetFlow architektura, na které funguje vlastní protokol, se typicky skládá z jednoho nebo několika exportérů sbírajících NetFlow data a jednoho kolektoru, který tato data ukládá. Existují dva typy NetFlow architektury, které se ve své podstatě pouze mírně odlišují použitou technologií a umístěním exportéru.

3.4.1 Exportér

NetFlow exportér je síťový prvek sbírající NetFlow data z linky, ke které je připojen, a posílající je k uložení v příslušném kolektoru. K odeslání dat dochází ve dvou případech. Prvním případem je ukončení spojení mezi zdrojovou a cílovou lokací, tedy ukončení flow. Definice samotného ukončení záleží na protokolu, pomocí kterého je spojení provedeno. Druhým případem, kdy dojde k odeslání dat, je vypršení nastaveného časovače v exportéru, typicky pět minut [35]. K tomuto dochází při tocích s velkým časovým intervalem trvání. Sledování pak pokračuje jako nový tok, který má svůj počátek ve stejném časovém okamžiku jako předchozí tok své ukončení.

3.4.2 Kolektor

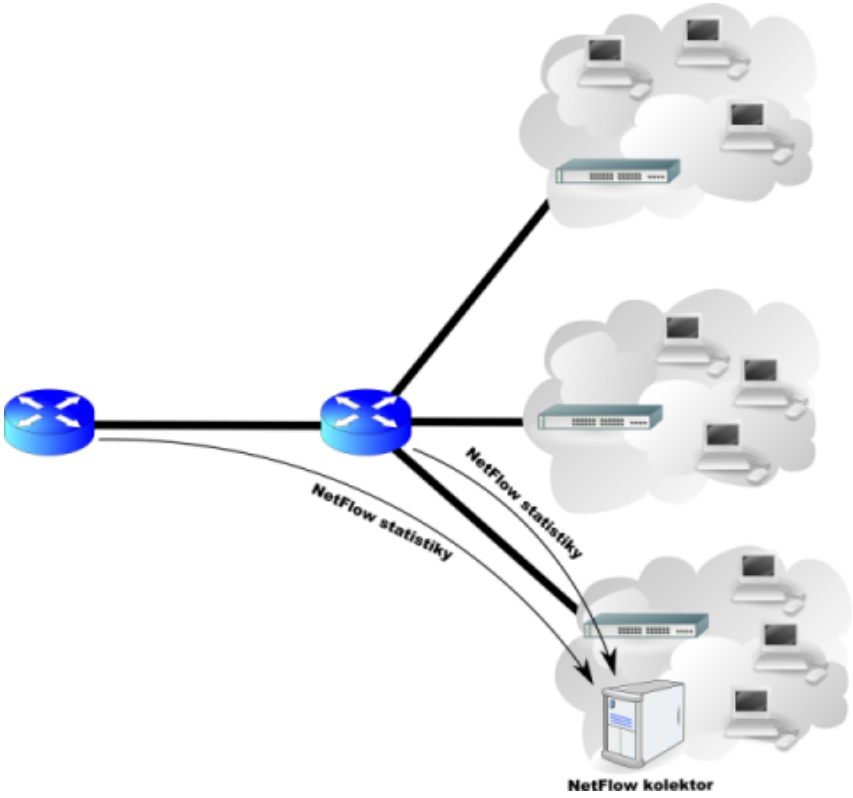
Termínem kolektor je v NetFlow architektuře označení pro zařízení s velkou diskovou kapacitou, kam jsou ukládána data ze všech přidělených NetFlow exportérů v síti. K tomuto datovému úložišti má většinou přístup některá z aplikací umožňující běžně potřebné operace nad daty, jako je filtrování toků dle filtračních pravidel, jejich agregace dle zadaných kritérií a zobrazování výstupu do příkazového řádku, případně pomocí webového rozhraní.

3.4.3 Tradiční architektura

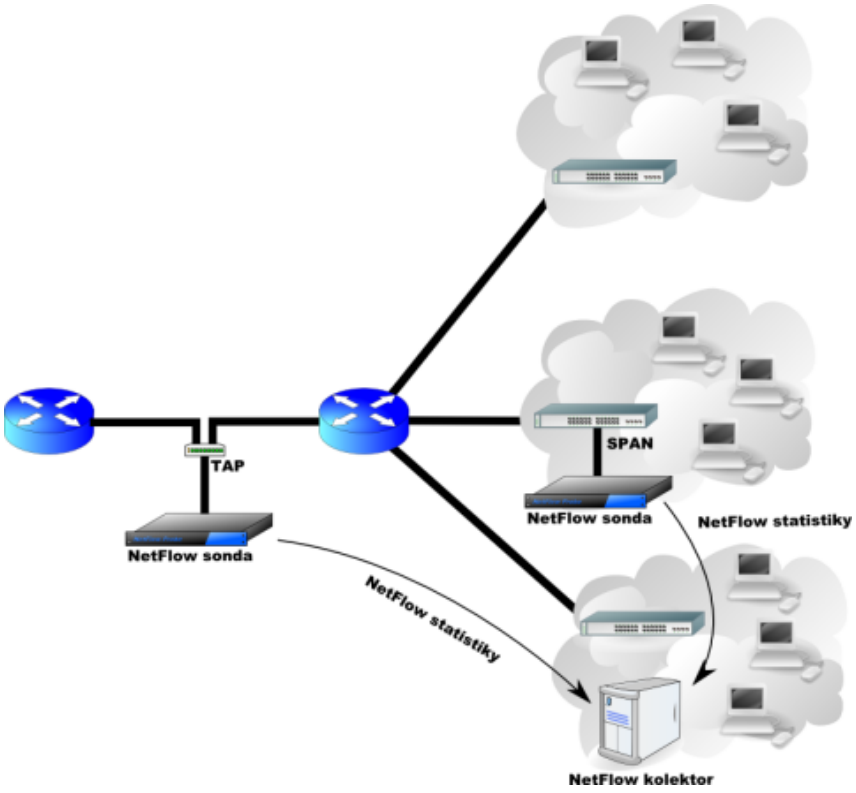
Jak již bylo dříve uvedeno, technologie NetFlow vznikla jako doplňková služba pro směrovače firmy Cisco Systems. Proto se v tradiční architektuře počítá s přítomností exportéru přímo ve směrovačích v konkrétní síti. Toto řešení má ovšem několik zásadních nevýhod. Zařízení musejí mít vysoký výkon, aby zvládla kromě své primární funkce i sběr a export NetFlow dat do kolektoru. Proto se zde často používá vzorkování, aby došlo k odlehčení zařízení. Snižuje se tedy přesnost dat a tím i pravděpodobnost odhalení incidentu. Na konec je potřeba zmínit, že tato zařízení jsou poměrně drahá, což snižuje jejich dostupnost.

3.4.4 Moderní architektura

Druhým přístupem, který je dnes uplatňován častěji, je takzvaná moderní architektura. Klíčovou myšlenkou této architektury je použití nezávislých hardwarových prvků (sond) připojených do monitorované sítě. Většinou jsou umístěny v klíčových uzlech, jako je například místo připojení sítě do internetu. Tyto sondy se chovají k datům při jejich průchodu transparentním způsobem, toky v síti pouze monitorují a zaznamenávají. Žádným způsobem do nich tedy nezasahují. NetFlow data jsou pak odesílána na patřičný kolektor pomocí dedikované linky a nemají tak na provoz žádný zkreslující vliv. Mezi výhody tohoto přístupu patří vyšší výkonnost sond a také nižší pravděpodobnost jejich odhalení a odstavení případným útokem. V neposlední řadě je třeba zmínit nižší cenu hardwarových sond oproti exportérům integrovaným do směrovačů.



Obrázek 3.1: Tradiční architektura [34].



Obrázek 3.2: Moderní architektura [34].

Kapitola 4

Statistické metody detekce v síťovém provozu

Statistické metody, jak již jejich název napovídá, jsou založeny na sledování statistických veličin získaných na základě dostupných NetFlow dat popisujících chování jednotlivých strojů v monitorované síti. Ve své podstatě jsou sledovány odchylky v chování stroje od normálního stavu, které značí pravděpodobnou anomálii. Na typu a propracovanosti metody pak záleží přesnost detekce, počet falešných poplachů a další veličiny.

Kromě statistických metod detekce v síťovém provozu je ve spojitosti s NetFlow také používáno detekce pomocí detekčních pravidel [26] a rozhodovacích stromů [28], které ovšem nejsou v této práci zahrnuty. V následujících odstavcích této kapitoly jsou podrobněji představeny všechny statistické metody použité v této práci. Podrobnější popis jejich implementace se pak nachází v příslušné kapitole.

4.1 Klouzavé průměry

Ve své podstatě se jedná o typ filtru s konečnou impulzní odezvou, který je používán k analýze množiny dat a určení jejich trendu pomocí tvorby série průměrů rozdílných podmnožin celkové množiny všech dat. Jedná se o příklad dolní propusti používané ve zpracování signálu, která odstraňuje vyšší frekvence. Matematickým pohledem je pak klouzavý průměr typem konvoluce.

Jedno z využití klouzavého průměru společně s časovými řadami je zmenšení krátkodobých fluktuací a zvýraznění dlouhodobých trendů, či cyklů v datech. Práh mezi krátkodobostí a dlouhodobostí je nastavován pomocí velikosti klouzajícího okna a případně vah u jednotlivých položek váženého klouzavého průměru. Vzhledem ke svým vlastnostem jsou klouzavé průměry používány k analýzám burzovních dat a dalších ekonomických ukazatelů.

4.1.1 Prostý klouzavý průměr

Prostý klouzavý průměr (SMA) je nejjednodušší metodou výpočtu hodnoty z časové řady. V jistém smyslu vyjadřuje typickou hodnotu popisující soubor velkého počtu dat. Počítaná hodnota S_t je získána v čase t sečtením všech předcházejících hodnot x a vydělením jejich součtu počtem hodnot k, pro které vždy platí, že $k \geq 1$. Se zvyšující se velikostí k je kladen větší význam na data z minulosti a tedy dlouhodobý trend. V případě menšího k je obecně kladen větší důraz na aktuální data. Výše uvedené je možné vyjádřit formálně [16]:

$$S_t(k) = \frac{1}{k}\sum_{i=1}^{k} x_{k-i} = \frac{x_t + x_{t-1} + x_{t-2} + \ldots + x_{t-k}}{k} \tag{4.1}$$

4.1.2 Vážený klouzavý průměr

Vážený klouzavý průměr (WMA) je zobecněním předcházejícího prostého klouzavého průměru. Jeho síla je v tom, že je možné přiřadit každému prvku x_i jeho váhu w_i, která je nezáporným číslem, a tak zdůraznit nebo potlačit jeho důležitost v rámci hodnot, ze kterých je průměr počítán. V tomto konkrétním případě použitém v prací se jedná o normalizovaný vážený průměr [16], jehož součet všech vah jednotlivých prvků prvků použitých k výpočtu průměru musí být roven hodnotě 1 [16], tedy:

$$\sum_{i=1}^{k} w_i = 1 \tag{4.2}$$

Dále pak také platí jako u předchozího případu, že počítaná hodnota S_t je získána v čase t sečtením součinu předcházejících hodnot x s jejich váhou w. Opět zde platí, že musí být vždy $k \geq 1$. Celý výpočet průměru je potom vyjádřen pomocí:

$$S_t(k) = \sum_{i=1}^{k} w_{k-i}x_{k-i} = w_t x_t + w_{t-1}x_{t-1} + w_{t-2}x_{t-2} + \ldots + w_{t-k}x_{t-k} \tag{4.3}$$

4.2 Metoda ASTUTE

Detekční metoda A Short-Timescale Uncorrelated-Traffic Equilibrium (dále ASTUTE) kolektivu autorů Silveira, Diot, Taft, Govindan vychází z předpokladu, že pokud je určitý počet toků multiplexován na ne zcela vytížené lince, pak změny jejich velikosti v krátkých časových kamžicích inklinují k vzájemnému vyrušení a průměrná změna napříč těmito toky se blíží nule [24].

Tato rovnováha platí, pokud jsou toky téměř nezávislé a je porušena změnami v provozu způsobenými několika, potencionálně malými, korelovanými toky [24]. Mnoho anomálií, ať už jsou způsobeny útočníky nebo chybou konfigurace, v síťovém provozu splňuje tento popis [24]. Na základě uvedeného pozorování byla navržena metoda využívající všechny tyto principy k sestavení detekční metody pro korelované anomální toky.

ASTUTE má oproti jiným používaným metodám detekce tři zásadní výhody [24]. Za prvé nepotřebuje čas pro naučení se charakteristik provozu, takže je přirozeně imunní proti data-poisoningu. Dále je vzhledem ke své specializaci na anomálie vyznačující se silně korelovanými toky ASTUTE přesnější v jejich označování než jiné metody [24]. Za třetí jakmile dojde k označení anomálního časového okna, ASTUTE poskytuje informace o anomálii, které usnadňují její prozkoumání a zařazení.

4.2.1 Rovnovážný model

Nyní je třeba definovat potřebné pojmy, aby bylo možné uchopit jak rovnovážný model popisující metodu, tak popsat její důsledky a z nich vycházející návrh detektoru anomálií. Síťový tok ($flow$) je množina paketů sdílejících stejnou množinu vlastností (zdrojový a cílový port, zdrojová a cílová IP adresa). Pro studium vývoje toku je čas rozdělen na

časové intervaly o stejné velikosti nazývané (*bins*). Změny velikosti (*volume*) síťových toků f v průběhu časového okna i jsou označovány jako $x_{f,i}$ a jedná se o počet paketů, nebo Bytů v korespondujícím časovém okně.

V tomto modelu je průchod toků skrze bod pozorování v síti modelován jako diskrétní bodový proces [5]. Bodové procesy se používají k modelování náhodných událostí v čase a prostoru (nejčastěji dvou nebo třídimenzionálním) [2]. Jsou determinovány časovým okamžikem a souřadnicemi [5]. V tomto konkrétním případě se jedná o trvání toku a jeho velikost v každém časovém oknu. Každý flow je tedy unikátně definován těmito náhodnými proměnnými:

- s_f je časové okno, kdy se sledovaný tok objevil na lince
- d_f představuje počet časových oken, ve kterých je sledovaný tok aktivní
- $\vec{x_f} = (x_{f,s_f}, \ldots, x_{f,s_f+d_f-1})$ je vektorem s velikostmi toku pro každé časové okno, kdy je sledovaný tok aktivní

Protože model umožňuje jakékoliv rozložení příchozího i označujícího procesu [24], byly vysloveny dva následující předpoklady. První z nich říká:

Předpoklad 1 *Nezávislost toků - vlastnosti sledovaného toku* $(d_f, s_f, \vec{x_f})$ *jsou na vlastnostech všech ostatních toků nezávislé.*

Existují pouze dva známe předpoklady, kdy může být porušena nezávislost síťových toků. První je, že toky jsou uskupeny do session. Například pokud je otevřena v prohlížeči webová stránka, pak se klient připojuje na mnoho dalších serverů, aby stáhl potřebný obsah stránky jako jsou obrázky a různé další součásti.

Druhý případ, kdy dochází ke korelaci síťových toků a tedy porušení prvního předpokladu nastává, pokud toky sdílí stejnou frontu v síťových zařízeních. K tomuto jevu dochází pouze pokud je linka plně vytížena a některé toky musí nuceně snížit svou propustnost, aby jiné mohly svoji propustnost zase zvýšit.

Předchozí práce ukázaly, že tyto důvody pro korelaci v reálném provozu nastávají velmi zřídka [24]. Jedním z důvodů je, že páteřní síťové linky jsou již z principu svého návrhu podstatně předimenzované a z tohoto důvodu u nich nedochází k jejich plnému využití [24].

Předpoklad 2 *Stacionarita - distribuce procesu příchodu toků a jejich značení se v průběhu času nikdy nezmění.*

Stacionarita záleží na velikosti časového okna (*binu*), ve kterém jsou síťové toky pozorovány [24]. Síťové toky mohou vykazovat nestacionaritu při časových oknech delších než pět minut, protože se do pozorování promítají denní a týdenní cykly, státní svátky a další dlouhodobé trendy. Při všech časových oknech, které jsou menší než jedna hodina, lze však síťový provoz dobře modelovat stacionárním procesem [24].

Jsou-li uvažována dvě po sobě následující časová okna označená i a $i+1$. Nechť $\mathcal{F}$ je množinou všech síťových toků, které jsou aktivní v časových oknech i a $i+1$. Pro tok $f \in \mathcal{F}$ potom platí, že $\delta_{f,i} = x_{f,i+1} - x_{f,i+1}$ je změna velikosti toku f mezi časovými okny i a $i+1$. Pokud sledovaný tok začíná svou existenci až v okně $i+1$, nebo je ukončen již v okně i, potom je hodnota $x_{f,i+1}$, nebo $x_{f,i}$ v $\delta_{f,i}$ rovna nule. Dále je také definována Δ_i jako množina všech $\delta_{f,i}$ pro každý tok $f \in \mathcal{F}$.

Níže uvedený teorém je úplným shrnutím hlavních důsledků výše představeného modelu a je základem dále prezentované metody pro detekci anomálií v síťovém provozu.

Teorém 1 *Pokud současně platí předpoklad 1 a zároveň platí předpoklad 2, pak hodnoty v Δ_i jsou nezávislé a shodně distribuované náhodné hodnoty s průměrem rovným nule. Pro dva naprosto náhodně zvolené toky f a g náležící do množiny všech toků tedy $\mathcal{F}$ platí:*

1. *$\delta_{f,i}$ má nulový průměr*
2. *pokud $f \neq g$, pak $\delta_{f,i}$ je nezávislé na $\delta_{g,i}$*
3. *$\delta_{f,i}$ a $\delta_{g,i}$ mají shodné rozložení pravděpodobnosti*

První bod teorému je možné dokázat pro libovolný tok $f \in \mathcal{F}$ následujícím způsobem. Podmínkou pro jeho platnost je $d_f = d$ a zároveň $\vec{x_f} = \vec{x} = (x_{s_f}, \ldots, x_{s_f+d-1})$. Pro pevně dané d se čas příchozího toku s_f může měnit od v rozmezí od $i-d+1$ do $i+1$. S ohledem na stacionaritu toků popsanou v 2 je pro f stejně pravděpodobný příchod v libovolném časovém okně v uvedeném rozsahu. Je pak tedy možné říci, že s_f splňuje podmínky rovnoměrného rozdělení mezi $i-d+1$ a $i+1$. Průměr $\delta_{f,i}$ můžeme poté vyjádřit jako podmínku přes všechny možné hodnoty s_f a pro každou s pravděpodobností $\frac{1}{d+1}$:

$$\sum_{s_f=i-d+1}^{i+1} \frac{\delta_{f,i}}{d+1} = \sum_{i=s_f-i}^{s_f+d-1} \frac{x_{i+1}-x_i}{d+1} = \frac{x_{s_f+d}-x_{s_f-1}}{d+1} = 0 \tag{4.4}$$

V posledním kroku výše uvedeného výrazu 4.4 je užito již dříve zmíněného faktu, že x_{s_f+d} a x_{s_f-1} jsou nulové, pokud tok f začíná v časovém okně s_f a končí v okně s_f+d-1.

Druhý bod teorému je přímým důsledkem předpokladu 1. Pokud jsou $\vec{x_f}$ a $\vec{x_g}$ na sobě nezávislé pro dva rozdílné toky f a g náležící do $\mathcal{F}$, pak změny jejich velikosti $\delta_{f,i}$ a $\delta_{g,i}$ mezi časovými okny i a $i+1$ jsou na sobě také nezávislé.

Poslední bod je důsledkem předpokladu 2 a dříve uvedeného pozorování, že podmíněné rozložení $\delta_{f,i}$ dané d_f a $\vec{x_f}$ závisí pouze na d_f a $\vec{x_f}$. Vzhledem k tomuto pozorování platí, že rozložení d_f a $\vec{x_f}$ není závislé na f. Okrajové rozložení $\delta_{g,i}$ je pak také nezávislé na f.

4.2.2 Detektor založený na metodě ASTUTE

Na důsledcích uvedeného teorému je založen dále popisovaný detektor silně korelovaných síťových toků, jehož Nulová hypotéza je důsledkem Teorému 1. Nechť F jsou všechny toky aktivní v časovém okně i se změnou velikosti určenou $\delta_{f,i}$. Potom $\hat{\delta}_i$ je výběrový průměr a $\hat{\sigma}_i$ je výběrová směrodatná odchylka změny objemu toků:

$$\hat{\delta}_i = \sum_{f=1}^{F} \frac{\delta_{f,i}}{F} \;\therefore\; \hat{\sigma}_i = \sqrt{\sum_{f=1}^{F} \frac{(\delta_{f,i}-\hat{\delta}_i)^2}{F-1}} \tag{4.5}$$

Pokud platí Teorém 1 pro velká F, pak $\hat{\delta_{f,i}}$ má interval spolehlivosti (1-p), který je dán Centrální limitní větou, kde hodnota $K(p)$ je percentilem $1-p/2$ Gaussova rozdělení pravděpodobnosti:

$$I_{\delta_i} = \left[\delta_i - K(p)\frac{\hat{\sigma}_i}{\sqrt{F}}\,,\; \delta_i + K(p)\frac{\hat{\sigma}_i}{\sqrt{F}}\right] \tag{4.6}$$

Tok tedy splňuje ASTUTE, pokud $I_{\hat{\delta}_i}$ obsahuje 0. V opačném případě se jedná o anomálii v časovém okně i. Je tedy zřejmé, že účinnost algoritmu závisí na volbě $K(p)$. Pokud je zvětšována úroveň spolehlivosti $1-p$, pak je také zvětšován interval spolehlivosti $I_{\hat{\delta}_i}$. Pro

danou množinu toků je interval charakterizován hodnotou $K(p)$ 4.6. Nejmenší hodnota $K(p)$, aby interval obsahoval 0 je:

$$K' = \frac{\hat{\delta}_i}{\hat{\sigma}_i}\sqrt{F} \tag{4.7}$$

Hodnota K' se nazývá ASTUTE assessment value (AAV) časového okna. ASTUTE je porušena právě tehdy, když platí $|K'| > K(p)$ [24]. Algoritmus výše popsaného detektoru tedy bude vypadat následovně:

Algorithm 1 Zjištění anomálie v časovém okně pomocí detektoru ASTUTE

Vstup: Všechna časová okna $\wedge$ treshold $K(p)$ určující false positive hodnotu p.
Výstup: Časové okno obsahuje anomálii. $\vee$ Časové okno neobsahuje anomálii.

```
for all dvě sousední časová okna i a i − 1 do
    for all f ∈ F do
        Vypočti změnu velikosti δ_{g,i} mezi dvěma časovými okny.
    end for
    Vypočítej AAV (K') dle rovnice 4.7.
    if |K'| > K(p) then
        return Časové okno i obsahuje anomálii.
    else
        return Časové okno i neobsahuje anomálii.
    end if
end for
```

Pro zajištění dobrých detekčních vlastností je AAV vypočítáno vždy pro šest rozdílných agregací prohledávaných toků. Prvním způsobem je *cílová IP, zdrojová IP, srcport, dstport, protokol*. Druhý agreguje dle *zdrojová IP, cílová IP*. Třetí používá k agregaci *zdrojová IP* a čtvrtý *cílová IP*. Poslední dva způsoby jsou *zdrojový port* a *cílový port*.

Nicméně je možné použít jakékoliv jiné kombinace pětice definující síťový tok, či jiné dostupné položky z NetFlow pro dosažení potřebného výsledku [24]. Rozdílné způsoby agregace jsou zvoleny, protože poskytují větší spolehlivost detektoru a za druhé umožňují odhalit anomálie, které lépe vyniknou při určitém způsobu agregace. Síťový tok jako takový je považován za anomální, pokud byl alespoň při jednom způsobu agregace označen vztahem $|K'| > K(p)$ jako anomální.

Z proběhlých pozorování bylo odvozeno, že k porušení stacionarity metody ASTUTE dochází při časových oknech delších než 15 minut. Dále je však používáno časové okno o velikosti 5 minut [24]. Pětiminutové intervaly sběru dat jsou základní volbou u drtivé většiny sond zpracovávajících NetFlow data [35].

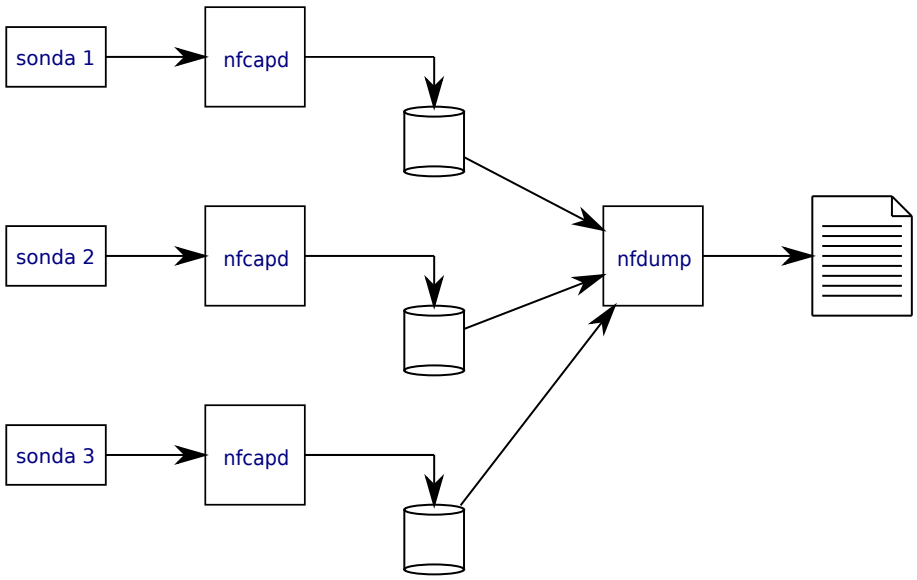
Kapitola 5

Zvolené implementační nástroje

V následující kapitole jsou popsány nástroje použité v implementační části této práce. Důraz je kladen na popis dvojice nástrojů Nfdump a NfSen. Také je popsán nástroj zde RRDTool používaný v NfSenu k uložení a vykresování dat. Dále jsou pak popsané podpůrné implementační nástroje SQLite, Perl Data Language a jqPlot.

5.1 Nfdump

Nfdump je souborem nástrojů určeným pro sběr a zpracování NetFlow dat verze 5, verze 7 a verze 9. Je distribuován pod BSD licencí a je spustitelný na všech BSD a Posix platformách [35]. Cílem tohoto souboru nástrojů je, aby bylo možné snadno a rychle analyzovat NetFlow data z minulosti, stejně jako možnost nepřetržitě monitorovat aktuální zajímavé toky. Počet síťových toků z minulosti uložených zpětně do minulosti je limitován pouze velikostí dostupného diskového prostoru pro NetFlow data. Nástroje jsou napsány v C a jsou optimalizovány pro efektivní filtrování.



Obrázek 5.1: Obecná architektura souboru nástrojů.

Všechna získaná data jsou ukládána před zpracováním na disk do adresářové struktury, skládající se z jména kanálu a podadresářů ve struktuře `YYYY/MM/dd` v pravidelných, typicky 5 minutových, intervalech do souborů s názvem `nfcapd` a koncovkou v podobě časové známky ve formátu `YYYYMMddhhmm`, která označuje, že soubor obsahuje data od tohoto okamžiku dále [35]. Je tak zajištěno oddělení ukládání dat od jejich analýzy pro případ selhání. Analýzu dat je pak možné provádět jak pro jeden soubor, tak i pro několik souborů z různých adresářů najednou. Výstup lze uskutečnit v textové formě nebo pomocí binárních dat uložených do souboru, které je poté možné znovu zpracovat. Z množiny všech nástrojů balíku nfdump jsou pro tuto práci nejvýznamnější následující dva programy.

5.1.1 Nfcapd

Démon nfcapd čte data ze sítě a ukládá je do souborů označených časovým razítkem, které pravidelně rotuje v pětiminutových intervalech. Program nfcapd čte NetFlow data verze 5, 7 a verze 9. Pro každý NetFlow stream, označovaný také jako kanál, je potřeba mít jednoho aktivního nfcapd démona.

5.1.2 Nfdump

Program nfdump čte data uložená do již dříve zmíněné adresářové struktury programem nfcapd. Pomocí něj lze vyfiltrovat a agregovat síťové toky odpovídající vytvořeným filtrům popisujícím rozličná kritéria jako například cílová nebo zdrojová IP adresa (verze 4 nebo 5), počet toků ve flow a mnoho dalších. Celkový přehled všech položek použitelných pro tvorbu filtrů, agregaci a také správný formát zápisu je možné dohledat v manuálových stránkách programu nfdump nebo v [35].

Dále je možné generovat statistiky nejvýraznějších toků (*TopN*) s ohledem na některou z přednastavených možností. Typicky se jedná o IP adresu, port, atd. Je možné si vybrat některý z přednastavených formátů výstupů dat, nebo si vytvořit formát vlastní. Formáty výstupu poskytují různé úrovně detailu informací a některé jako csv, či pipe jsou obzvláště vhodné pro další automatické zpracování.

5.2 RRDTool

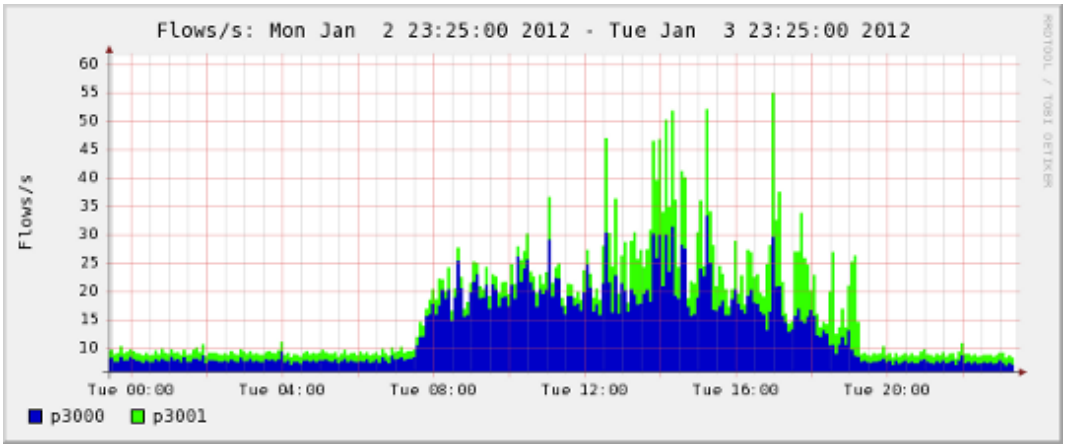
Nástroj RRDTool je otevřeným průmyslovým standardem pro vysoce výkonné ukládání a vykreslování časově závislých dat a může být jednoduše integrován ve skriptech napsaných v shellu, pythonu, perlu, ruby a dalších jazycích [20]. Pracuje na principu round robin. Jinými slovy řečeno po dosažení maximálního možného počtu hodnot je další hodnota zapsána místo nejstarší hodnoty.

RRDTool předpokládá na vstupu přísun časově proměnných dat v určitém intervalu. Databáze je uložena v binárním souboru a skládá se z vlastních dat (DS) a round robin archivu (RRA). DS je možné vybrat z několika možných typů jako jsou například absolutní hodnota a čítač. DS má také přiřazen určitý počet RRA, která jsou aktualizována při příchodu nové hodnoty do databáze. Interval mezi příchodem dvou hodnot je pak obvykle nazýván krok a je specifikován při vytvoření databáze. Dále již nemůže být měněn. Protože data nemusí být vždy dostupná ve správný okamžik, RRDTool automaticky interpoluje přijatá data, aby získal hodnotu (PDP) vyhovující nastaveným časovým krokům. Tyto hodnoty mohou být konsolidovány pomocí konsolidační funkce (CF), typicky se jedná o minimum, maximum, nebo průměr, aby vytvořily novou konsolidovanou hodnotu (CDP).

Po svém získání je tato konsolidované hodnota uložena do round robin archivu, který pokrývá celý časový úsek 5.1. Archiv kromě ukládání pevného počtu CDP, určuje kolik PDP bude použito pro výpočet a s jakou funkcí CF.

$$\textit{čas pokrytý } RRA = \textit{počet uložených } CDP * \textit{počet } PDP \textit{ na jeden } CDP * \textit{kroky} \quad (5.1)$$

RRDTool však neumí data pouze zapisovat. Poskytuje velmi silný nástroj pro generování grafů přímo ze své databáze. Také je možné vybírat různá RRA z jednotlivých databází a kreslit je do jednoho grafu. Mezi další užitečné vlastnosti patří předzpracování dat pomocí základních matematických operací a řídících struktur programu.



Obrázek 5.2: Příklad grafu vykresleného pomocí RRDTool.

5.3 NfSen

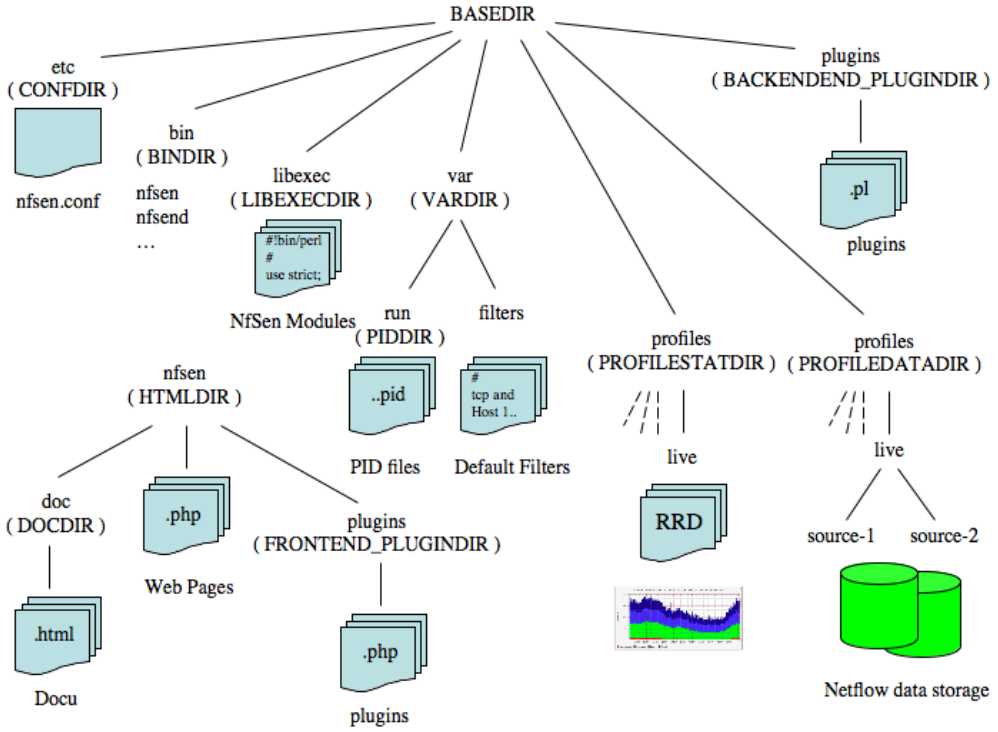
Pod názvem NfSen se skrývá open-source nástavba nad balíkem programů Nfdump a určená pro snadnější práci s NetFlow daty [36]. Hlavním cílem při jeho návrhu bylo vytvoření dostatečně jednoduchého rozhraní pro práci s rozsáhlými databázemi NetFlow záznamů, které by bylo dostupné pomocí webového prohlížeče. Velkou výhodou tohoto systému je jeho rozšiřitelnost pomocí pluginů.

5.3.1 Architektura

Celý NfSen je poměrně složitou aplikací navíc spolupracující s externími programy RRDTool a Nfdump. Je napsán v programovacích jazycích Perl a PHP. Části psané v jazyce PHP se starají o uživatelské rozhraní, odesílání dat a nastavení zapsaných uživatelem a také o vykreslovaní získaných dat. Jsou zde zobrazovány grafy generované frontendem prostřednictvím RRDTool. Umožňuje jednoduše sestavit dotaz pro Nfdump a poslat mu jej pomocí backendu. Zpět obdrží výpis toku odpovídající zadání. Části NfSenu napsané v programovacím jazyce Perl se starají o komunikaci s RRD databází. Spouští se zde příkazy pro ukládání dat a také příkazy pro generování grafů pro frontend. Dále se zde volá nfdump pro prohledávání a filtrování NetFlow dat.

Adresářová struktura Nfsenu se dělí na dvě významné části a je zachycena na obrázku 5.3. První je adresář pro frontend systému a jeho pluginů, kde jsou umístěny soubory

všechny soubory potřebné pro jejich vykreslování a navíc poněkud skrytá dokumentace celého systému NfSen. V druhém, hlavním, adresáři jsou pak mimo konfiguračních informací, modulů jazyka Perl a spouštěcích souborů dostupné RRD databáze se statistikami provozu ze všech profilů a jejich kanálů (PROFILESTATDIR) a také vlastní NetFlow data ukládaná ze sond (PROFILEDATADIR). Uložení dat respektuje již dříve uvedenou metodiku ukládání programu Nfdump a umožňuje tak snadnou spolupráci obou programů.



Obrázek 5.3: Adresářová struktura nástroje NfSen [36].

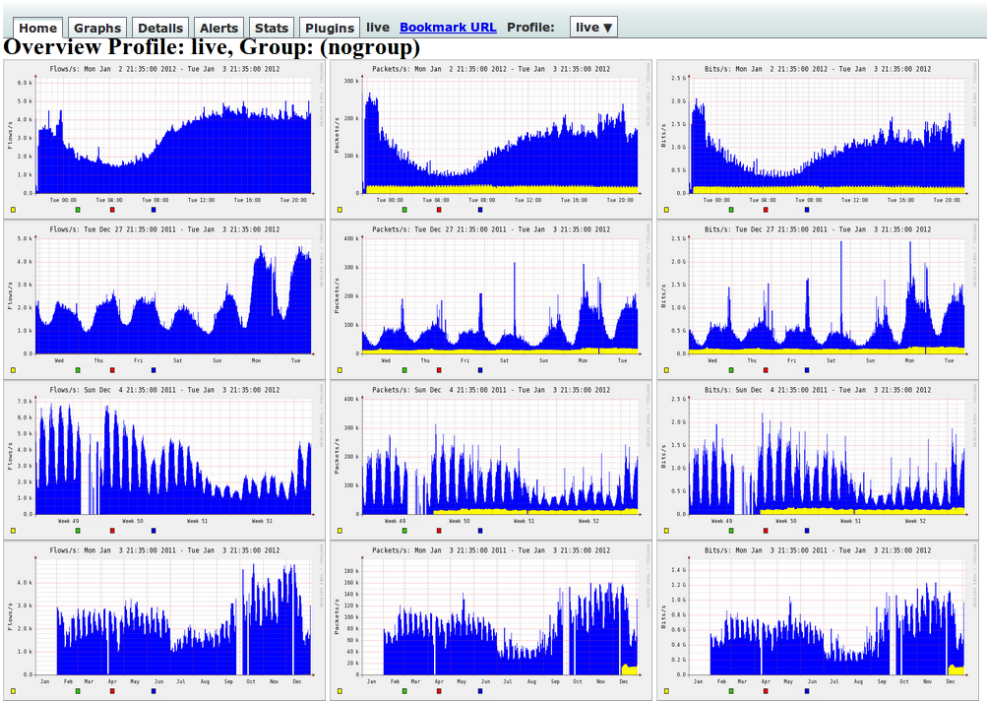
5.3.2 Běh aplikace

NfSen periodicky každých 5 minut kontroluje nové NetFlow záznamy v adresářové struktuře. Tyto záznamy jsou poté zpracovány v RRD databázi a jsou z nich vytvořeny potřebné statistické informace o počtu toků, paketů a síle provozu za poslední časové okno. Následně jsou vygenerovány patřičné grafy pro frontend. Z toho plyne, že je možné zjišťovat informace o síťových tocích s maximálně pětiminutovým zpožděním. Pětiminutová perioda zpracování také ovšem znamená, že běh všech aktuálně aktivních pluginů musí být ukončen před uplynutím této lhůty. Jinak bude docházet k hromadění úloh, které NfSen nestíhá zpracovat a může dojít až k zablokování celé aplikace narůstajícími nevyřízenými úlohami.

5.3.3 Uživatelské rozhraní a dostupné funkce

Webové rozhraní programu je poměrně povedené a umožňuje dobrý přehled o provozu v síti pomocí grafů generovaných pomocí RRDTool. Pokud uživatel chce využít detailnější náhled do provozu, pak mu NfSen umožňuje snadnou tvorbu filtrů pro hledání požadovaných toků a tvorbu *TopN* statistik. Tyto jsou zapisovány v syntaxi používané programem Nfdump

a je možné ty nejčastěji používané ukládat pro další použití. Formát výstup je pak možné konfigurovat shodně jako u programu Nfdump.



Obrázek 5.4: Úvodní stránka nástroje NfSen.

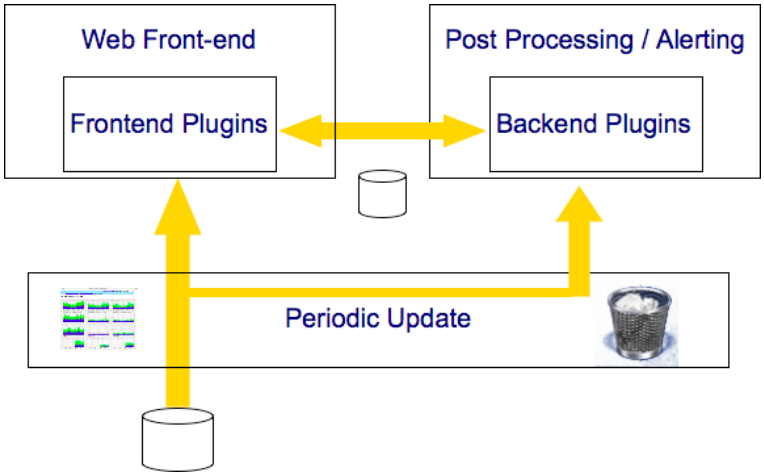
Kromě již zmiňovaného filtrování dat a tvorby *TopN* statistik poskytuje NfSen také rozhraní pro nastavení a zprávu takzvaných Alertů. Alerty slouží, jak již jejich název napovídá, k upozornění uživatelů na předem nastavené události, které NfSen zaznamená. Je možné u nich nastavit na které profily budou aktivovány. Jestli spouštěcí podmínka závisí na datech zjištěných samotným NfSenem, nebo na akci, či informaci některého z jeho pluginů. Reakce na událost může být odesláním předformátovaného emailu, nebo spuštění funkce v pluginu.

V neposlední řadě NfSen umožňuje také tvorbu a správu profilů a jejich kanálů. Je zde nastavována maximální velikost celého profilu. Také je možné zde konfigurovat filtrační pravidla společně se zdrojem dat pro jednotlivé kanály.

5.3.4 Pluginy

Jednou z nejdůležitějších vlastností NfSenu je jeho podpora pluginů. Umožňuje tak uživatelům podstatně rozšířit schopnosti tohoto nástroje, jak o vlastní výtvory tak o některé pluginy třetích stran. Malou knihovnu volně dostupných pluginů je možné nalézt například na [32] a pro vlastní tvorbu pluginu dávají autoři NfSen poměrně dobrý tutoriál na své webové stránce [36].

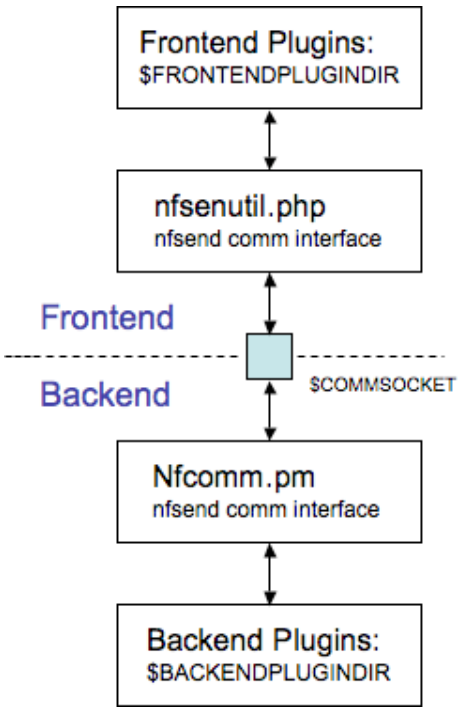
Všechny pluginy stojí na stejném principu backendu (backend plugin) a frontendu (frontend plugin) jako celý systém NfSen. Backend plugin je připojen k hlavnímu procesu NfSenu při jeho startu [36] a přidává do něj požadovanou funkcionalitu. Může se jednat o periodické zpracovávání dat, či implementaci upozorňujících podmínek a funkcí pro reakci na tyto podmínky. Frontendový plugin zobrazuje jakékoliv výstupy zaslané backendem a vzhledem



Obrázek 5.5: Koncept pluginu [36].

k tomu, že je napsán v PHP, tak je možné jej upravovat při zachování určitých pravidel téměř jako běžnou webovou stránku včetně použití javascriptu, Ajax a jQuery. Struktura pluginu a komunikace mezi jednotlivými součástmi je zachycena na obrázku 5.5.

Přestože by mohla být data z backend pluginu zobrazena v prohlížeči přímo přes PHP, nejedná se však v žádném případě o doporučený postup. Budoucí verze NfSenu by měly podporovat běh frontend a backend pluginů na rozdílných strojích [36]. Proto byla navržena vzájemná komunikace pomocí modulů zvaných NfSen sokety. Tento proces je blíže popsán na obrázku 5.6. Pomocí těchto soketů lze vytvořit vyhrazený kanál mezi frontend a backend instancí pluginu Je tak možné přes ně odesílat skalární hodnotu, případně pole.



Obrázek 5.6: Komunikační soket pro pluginy [36].

5.4 Perl Data Language

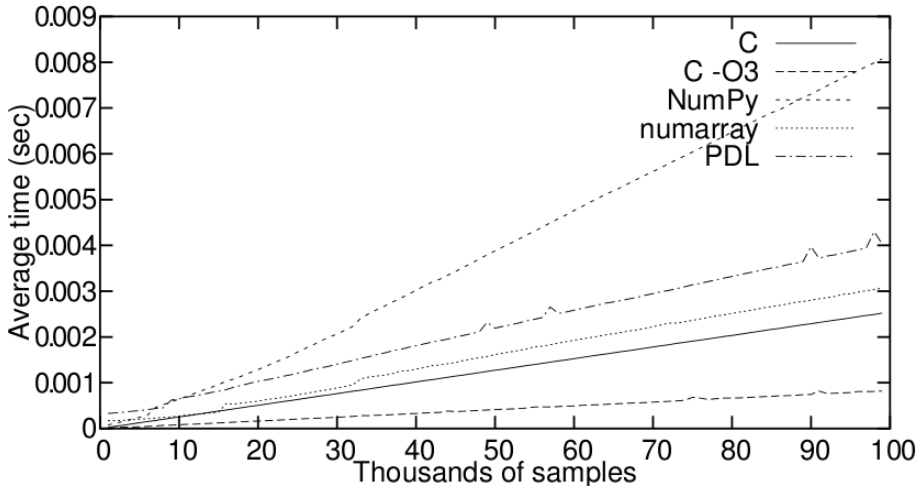
Vzhledem k předpokládané náročnosti výpočtu metody ASTUTE nad všemi toky s šesti rozdílnými způsoby agregace a s přihlédnutím k omezenému časovému úseku pro tento úkol, který je způsoben neustálým opakováním zpracovávání dat s relativně krátkou periodou 5 minut, bylo rozhodnuto využít k urychlení práce s velkým objemem NetFlow dat modul programovacího jazyka Perl s názvem Perl Data Language (PDL).

Tato skupina rozšíření umožňuje jazyku Perl pracovat extrémně rychle s velmi rozsáhlými N-dimenzionálními poli [14] díky implementaci modulů v jazycích C a Fortran [29]. Autoři projektu PDL na svých stránkách srovnávají její funkčnost a rychlost s komerčními produkty jako je IDL nebo Matlab [14], kterými jsou také moduly částečně inspirovány. Na stránkách PDL jsou tak dokonce pro uživatele dostupné i příručky pro migraci do prostředí PDL z Matlabu, či Scilabu.

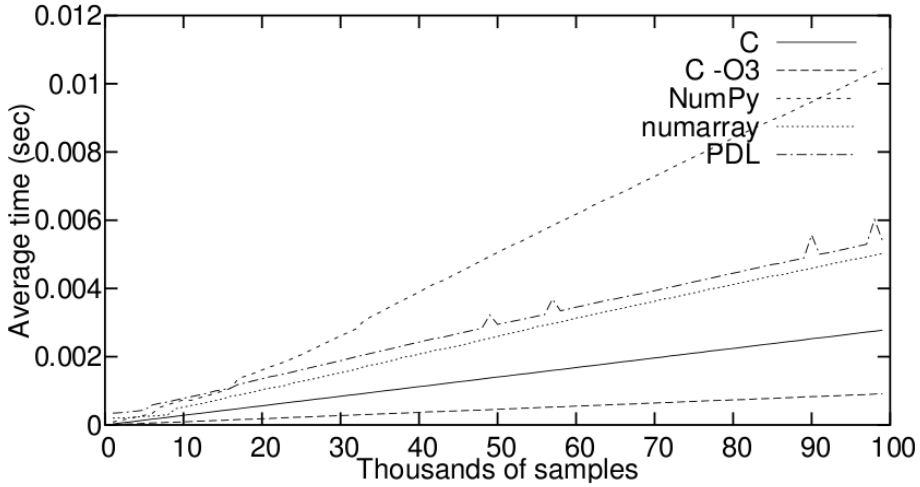
Základním stavebním prvkem celého PDL je takzvaný *Piddle* [14]. Jedná se o sérii čísel uspořádaných v N-dimenzionálním souboru dat. V prostředí Matlab odpovídá této struktuře nejblíže matice nebo vektor. Piddly jsou tím, co umožňuje efektivní ukládání a rychlý výpočet N-dimenzionálních matic. Jsou totiž silně optimalizované pro výpočty nad maticemi a vektory [14].

K práci s vlastními Piddly je dostupná celá řada metod od těch nejjednodušších, umožňujících přístup k samotným prvkům a vybírání podsouborů dat, až po metody umožňující Furierovu transformaci, kartografické uplatnění PDL a zpracování obrazových vstupů. Dále PDL disponuje vlastním optimalizovaným rozhraním pro IO operace, API pro přístup k potřebným knihovnám v jazyce C a Fortran. V neposlední řadě také obsahuje modul pro tvorbu dvou a tří dimenzionálních grafů [14].

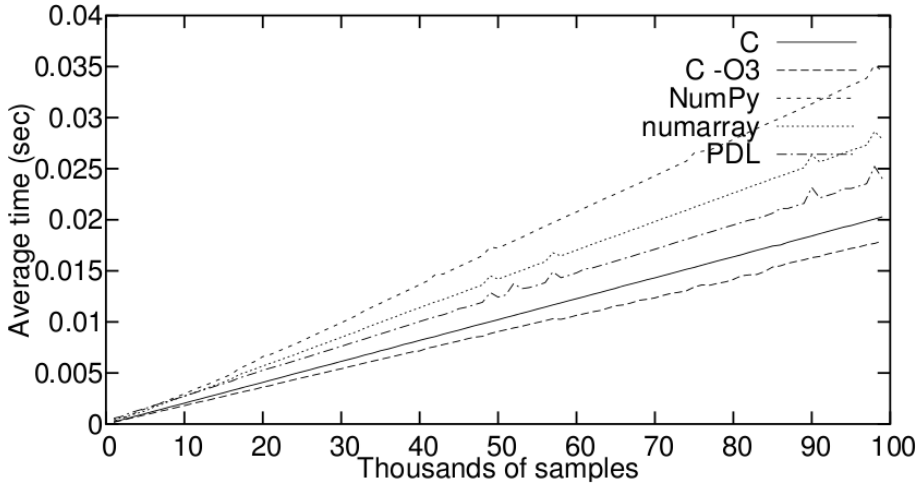
Článek A Peek on Numerical Programming in Perl and Python porovnává rychlost výpočtu za použití výše popisovaného modulu PDL s výpočtem pomocí implementace obdobného balíčku pro programovací jazyk Python (knihovna numPy) a s implementací totožného výpočtu v jazyce C přeloženou jak se zapnutou optimalizací kódu (-O3), tak bez optimalizace kódu. Jako referenční jsou také uvedeny implementace tohoto výpočtu v jazyce Perl a Python bez přidaných knihoven.



Obrázek 5.7: Test č. 1: numerická integrace $f(x) = x$ [7].



Obrázek 5.8: Test č. 2: numerická integrace $f(x) = x^2$ [7].



Obrázek 5.9: Test č. 3: numerická integrace $f(x) = \cos(x^2)\sin(x)$ [7].

Autor experimentu prováděl každý z těchto testů stokrát pro každou z kombinací hustoty zpracovávaných vzorků a programovacího jazyka. Z uvedených grafů je dobře patrné, že při použití Perlu společně s PDL je implementace obdobně efektivní jako implementace v jazyce C. Pokud je tedy možné řešený problém postavit jako práci s prvky v polích, jako v případě této práce, pak je Perl s modulem PDL vhodnou alternativou k tradičním přístupům v podobě programovacích jazyků C a C++ [7], což je ukázáno na grafech 5.7, 5.8 a 5.9 pocházejících z testování.

5.5 Databáze SQLite

SQLite je ultra lehkým SQL databázovým systémem naprogramovaným v jazyce ANSI-C [39] a neustále vyvíjeným od roku 2000 pod licencí public domain D. Richardem Hippem, který za něj obdržel v roce 2005 cenu Google O'Reilly Open Source Award [13].

Celá SQLite databáze je reprezentována jedním souborem na pevném disku. Toto umožňuje její použití i ve vestavěných zařízeních [39]. Aktuálně již podporuje téměř všechny

vlastnosti standardu SQL92, avšak v budoucnu se plánuje doplnění chybějících vlastností, mezi které patří `RIGHT OUTER JOIN` a `FULL OUTER JOIN`, kompletní podpora `ALTER TABLE`, kde momentálně není možné mazat a editovat sloupce a v neposlední řadě také úplná podpora triggerů a zápisu do pohledů [39].

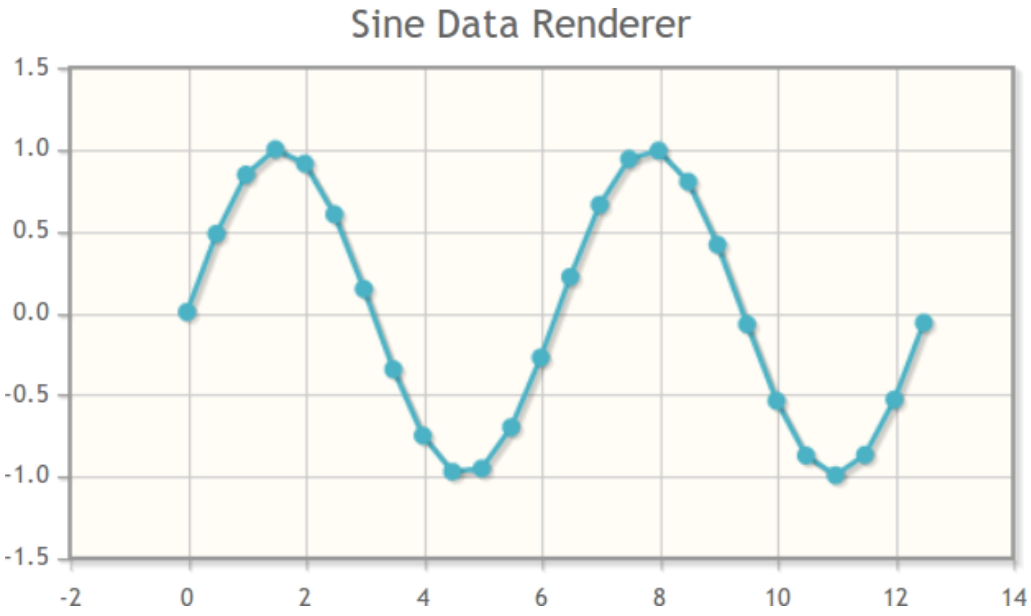
Mezi největší výhody databáze patří její samostatnost, protože nepotřebuje téměř žádnou podporu knihoven a operačního systému [39]. Dále je to bezserverovost a nepotřeba konfigurace. To znamená, že není potřeba serverových procesů a meziprocesní komunikace jako u jiných databázových enginů. S touto databází přistupují procesy přímo k datům uloženým v jednom souboru na pevném disku. Výhody tohoto přístupu jsou zřejmé. Není třeba instalovat server, nastavovat, inicializovat a spravovat jej. Nevýhody přístupu jsou pak následující: Není možné databázi chránit před chybami aplikace a zajistit lepší zamykání databáze a její lepší konkurentnost [39].

5.6 Knihovna jqPlot

JqPlot je poměrně novou knihovnou jazyka jQuery umožňující vykreslování sloupcových, čárových a koláčových grafů. Je dostupná pod licencemi MIT a GPL [9] a nedávno byla vydána její verze *1.0.0 beta2*.

Jak již z použití jQuery vyplývá, vykreslování grafů probíhá na straně klienta v prohlížeči. Server pouze poskytuje informace o datech pro vykreslení, popisu os, formátu grafu, legendě atd. Dochází tedy k přesunutí zátěže ze serveru na klienta. Navíc je možné grafy například v případě potřeby dynamicky překreslovat, zobrazovat aktuální hodnoty os na různých pozicích v grafu, přiblížit označenou oblast, atd. Také je možné grafy dále upravovat pomocí vlastních funkcí napsaných v javascriptu a jQuery.

Výhodou jqPlot je také, že je navrženo jako soustava pluginů [9]. V kódu je tedy možné použít pouze pluginy nezbytné pro vykreslení grafu a není nutné volat celou knihovnu jako takovou, což také přispívá ke snížení náročnosti na vykreslování.



Obrázek 5.10: Ukázka grafu vykresleného pomocí jqPlot [9].

Kapitola 6

Návrh a implementace statistických metod

Kapitola popisuje návrh a implementaci praktické části diplomové práce. Jedná se o pluginy systému NfSen, které implementují metody popsané v kapitole Statistické metody detekce v síťovém provozu. Tyto pluginy mohou sloužit ke zlepšení přehledu o sledované počítačové síti. Také umožňují sledovat nežádoucí jevy a upozorňují uživatele na jejich výskyt. V neposlední řadě slouží jako praktické otestování detekčních schopností metody ASTUTE uveřejněné v článku [24].

6.1 Návrh pluginu pro klouzavý průměr

Tento plugin má umožnit zainteresovaným osobám dohled nad počítačovou sítí pomocí této poměrně jednoduché metody a odhalovat tak anomálie projevující se změnou sledovaných oproti normálnímu stavu, vyjadřovaném v tomto případě průměrem s nastavenou odchylkou. Hlavní výhodou je jednoduchost, kdy uživatel nepotřebuje žádné detailní znalosti principů metody. Plugin by měl poskytovat následující funkcionalitu:

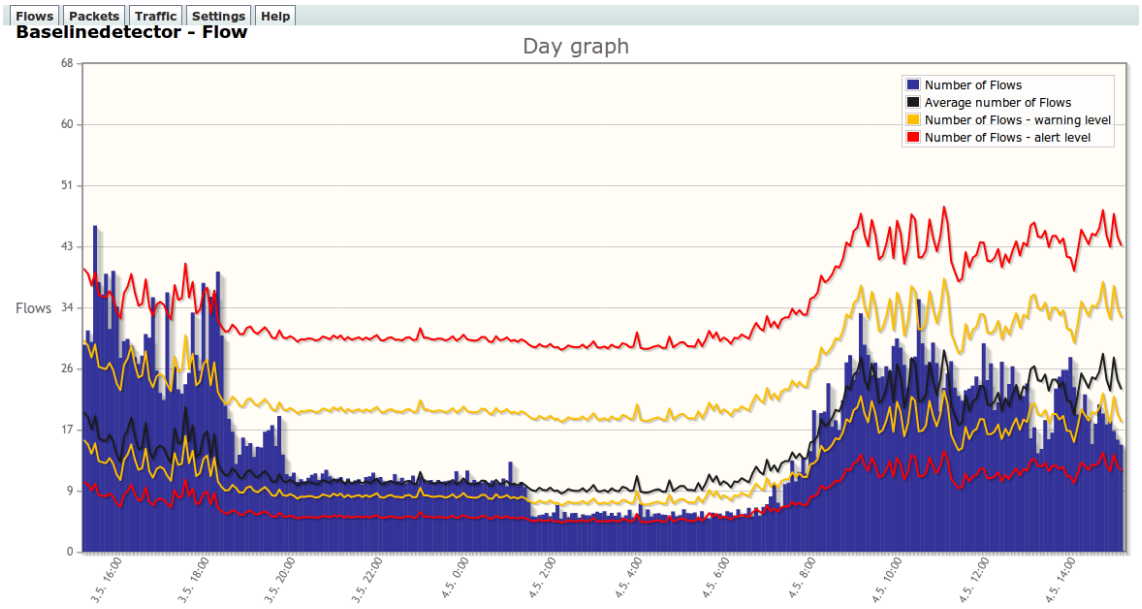
- Pracuje na všech profilech a jejich kanálech dostupných v programu NfSen.
- Umožňuje prohlížení hodnot toků, paketů a provozu v denní, týdenním a měsíčním časovém okně.
- Vykresluje průměrnou hodnotu a spodní i horní varovnou a kritickou linku (*Alert* a *Warning line*).
- Dovoluje zapínat a vypínat zobrazení vypočítaných linek v grafu.
- Obsahuje nastavení velikosti koeficientů pro každou jednotlivou linku.
- Lze zvolit mezi výpočtem linky jako procentuální části průměru nebo jako odchylky o určitou hodnotu.
- Ukládá průměry toků, paketů a provozu ve všech časových oknech.
- Umožňuje nastavit odeslání upozornění emailem, či provést jinou akci po překročení kritické hodnoty (*Alert line*).
- Dovoluje vymazat všechna data z vybraných tabulek.

6.2 Implementace pluginu pro klouzavý průměr

6.2.1 Frontend pluginu

Frontendová část pluginu implementujícího klouzavý průměr je rozdělena do pěti samostatných přehledných záložek. Jejich vzhled by částečně inspirován záložkou *Graphs* programu NfSen. Bylo však použito jiné technologie pro vykreslování grafů. Na rozdíl od programu NfSen, který používá grafy z RRDTool, jsou zde grafy vykreslovány pomocí knihovny jQuery a jejího pluginu pro vykreslování grafů jqPlot, který je sice náročnější na ovládnutí, ale produkuje graficky kvalitnější výstupy.

První tři záložky obsahují aktuální stavy toků (*Flows*), paketů (*Packets*) a provozu (*Traffic*). Každá z těchto záložek obsahuje tři grafy, které zobrazují příslušnou sledovanou veličinu v rámci dne, týdne a měsíce. Sledovaná veličina je v grafech pak zachycena pomocí sloupcových grafů modré barvy. Tyto grafy se, jak už bylo dříve řečeno liší nejen vykreslovaným časovým rozpětím, ale také velikostí jednotlivých časových oken. Změřená průměrná velikost sledované veličiny za určené období je pak vykreslena čárovým grafem černé barvy. Čárové grafy oranžové barvy zobrazují nastavenou spodní a horní varovnou hladinu (*Warning line*) pro daný průměr. Červené liniové grafy označují spodní a horní hladinu (*Alert line*), jejichž porušení značí poplach a v případě upozornění nastaveného v záložce *Alerts* v programu NfSen dojde k příslušné reakci.

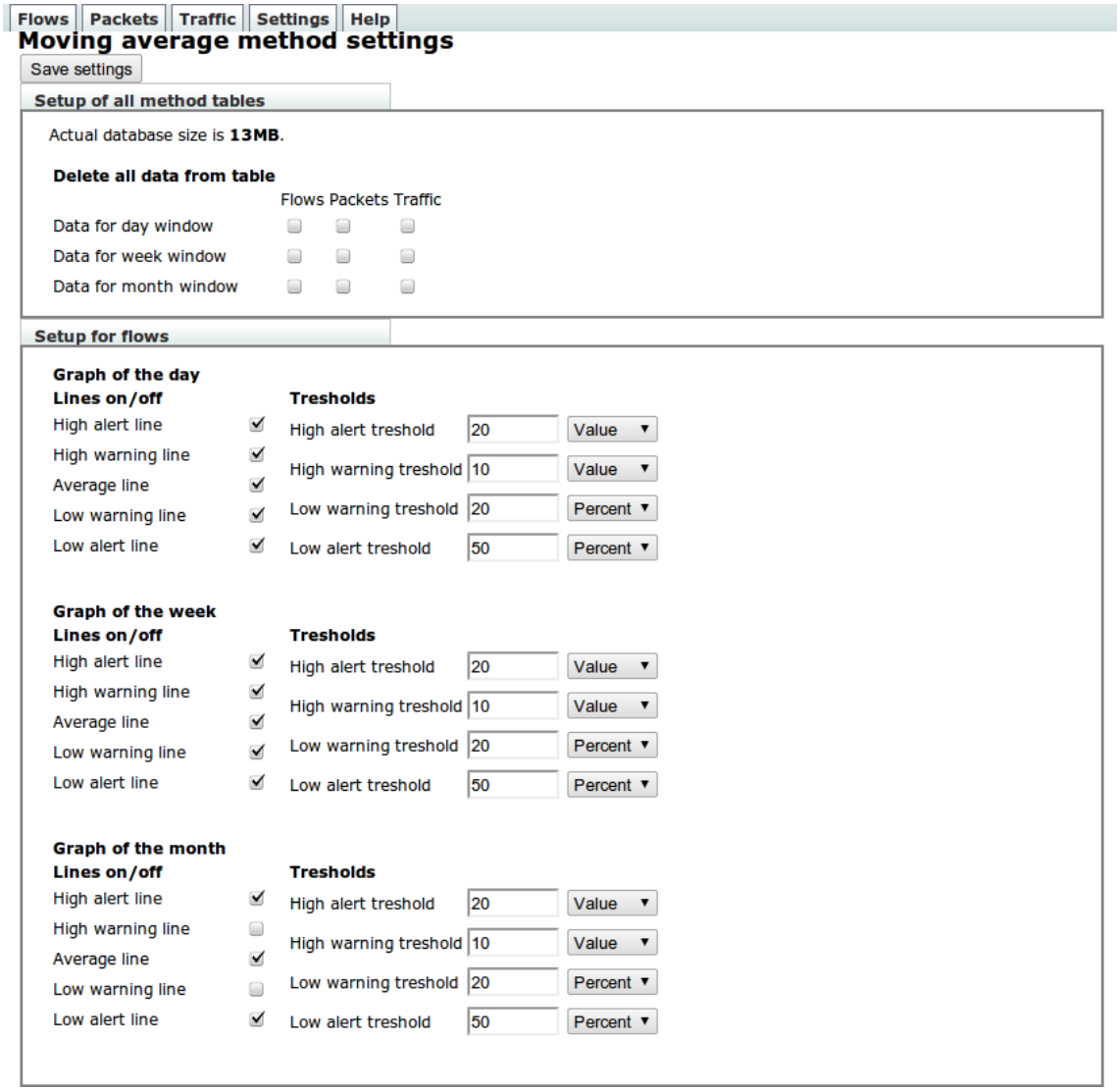


Obrázek 6.1: Graf ze záložky *Flows* zobrazující průměr toků v horizontu jednoho dne.

Čtvrtá záložka v pořadí, označená *Settings*, ukrývá veškerá nastavení pluginu a je rozdělena do čtyřech oblastí. První z nich obsahuje informace o aktuální velikosti celé databáze tohoto pluginu. Následuje tabulka se zatržítky, která umožňuje z libovolné tabulky vymazat veškerá data.

Dále následují tři obdobné oblasti. Každá z nich slouží pro nastavení potřebných údajů pro jednu z množiny veličin *Flows*, *Packets* a *Traffic*. Levá strana oblasti slouží pro zapínání a vypínání zobrazení jednotlivých liniových grafů označujících průměr, varování a poplach. Pravá strana oblasti je využita pro nastavování. Velikosti odchylek horních a dolních součástí grafů varování a poplachu. Volba se provádí zadáním číselné hodnoty do patřičného

políčka a poté vybráním, jestli se jedná o procentuální odchylku, či prostý posun o určitý počet hodnot od vypočteného průměru. Poslední záložka pak obsahuje nápovědu k jednotlivým akcím, které je možné v pluginu provádět.



Obrázek 6.2: Část záložky *Settings* obsahující nastavení tabulek pluginu a nastavení toků.

Vlastní nastavování poplachů v případě překročení nastavené odchylky se neděje přes záložku *Settings*, ale je kvůli standardizaci a přehlednosti realizováno přes vlastní rozhraní *Alerts* v programu NfSen. Zde je potřebné nastavit podmínky pro vyvolání poplachu dle tohoto pluginu. Následná akce pak může být libovolná. Plugin může odeslat pomocí emailu upozornění o proběhlém poplachu na nastavenou adresu, nebo může provést v podstatě libovolnou akci pomocí dalšího připraveného pluginu.

Hlavní funkcí pluginu, která se stará o vykreslení a také integraci pluginu do systému NfSen, je pevně definovaná funkce *run*, která se volá při každém načtení stránky.

6.2.2 Backend pluginu

Backend pluginu je možné rozdělit do tří zásadních oblastí. První z nich jsou vestavěné funkce nutné pro provázání pluginu s programem NfSen. Druhou oblast tvoří funkce volané dle potřeby z frontendové části pluginu. Poslední oblast tvoří funkce zajišťující podpůrné funkce, jako je komunikace s databází, výpočty a formátování výsledku.

Nejdůležitější funkcí z první skupiny je periodická funkce `run`. Ta je spuštěna pro každý z aktivních profilů v programu NfSen. Její průběh je zachycen v pseudokódu v algoritmu 6.2.2.

Algorithm 2 Periodický výpočet statistických dat.

Vstup: název NfSen profilu, název skupiny profilů
  Ověř existenci potřebných tabulek v databázi.
  Nastav parametry pro aktualizaci v 5 minutové periodě (denní časové okno).
  Získej data z RRD databáze NfSenu.
  Vypočti průměr a ulož jej do tabulky pro 5 minutovou periodu.
  if celá nebo polovina hodiny **then**
    Nastav parametry pro aktualizaci v 30 minutové periodě (týdenní časové okno).
    Získej data z RRD databáze NfSenu.
    Vypočti průměr a ulož jej do tabulky pro 30 minutovou periodu.
  end if
  if sudá hodina **then**
    Nastav parametry pro aktualizaci v 2 hodinové periodě (měsíční časové okno).
    Získej data z RRD databáze NfSenu.
    Vypočti průměr a ulož jej do tabulky pro 2 hodinovou periodu.
  end if
  Aktualizuj velikost databáze v tabulce *plugin_system_table*.

Nejdůležitější část funkce `run` je získání dat pro výpočet a výpočet průměru. Implementace výpočtu průměru je provedena standardním způsobem dle vzorce 4.1. Jednotlivé prvky pro výpočet však nejsou vybírány jako prostí předchůdci počítaného prvku. Pro časový okamžik i nejsou tudíž brány předcházející časové okamžiky $i-1$, $i-2$, atd. V této implementaci je zohledněna pro každé časové okno periodicita jeho dat. Prakticky to tedy znamená, že například pro týdenní časové okno jsou pro výpočet časového okamžiku *pondělí 08:00* brány jako předcházející hodnoty data z *pondělí 08:00* v předcházejících týdnech. Výhodou tohoto přístupu pak je přiblížení výpočtu reálné situaci, kdy mají na statistiku provozu vliv sezonní vlivy jako například státní svátky a dovolené [1].

Další důležitou periodicky volanou funkcí je `alert_condition`, která kontroluje splnění podmínek pro to, aby byl vyvolán poplach. Nejprve jsou z tabulky nastavení načteny hodnoty koeficientů pro všechny kontrolované hodnoty a časová období. Následně je pomocí funkce `checkAlert` provedena kontrola, zda počet toků, paketů nebo provozu spadá do rozmezí se spodní hranicí (*vypočtený průměr − odchylka daná koeficientem*) a horní hranicí (*vypočtený průměr + odchylka daná koeficientem*). Pokud tomu tak není, dochází k vyvolání poplachu a provedení nastavené akce.

Mezi funkce, které jsou volány z frontendu uživatelem patří `averageSettingsLoad`, `averageSettingsSave` a `getBaselineData`. Posledně jmenovaná funkce se stará o načítání dat pro jejich vykreslení v grafech frontendu. Na její vstup z něj přichází skrze soket informace o profilu, skupině do které patří, časové značce poslední aktualizace dat a dotazované hodnotě (*flow*, *packet*, *traffic*). Dle této hodnoty jsou na počátku nastaveny parametry pro

volání ostatních funkcí. Poté proběhne následující sekvence volání pro každý ze tří časových období. Získání dat z RRD databáze NfSenu. Načtení průměru z databáze pluginu. Výpočet odchylek pro kreslené liniové grafy ve frontendu. Vrácení naformátovaných dat skrze soket do frontendu.

Funkcí stojící za zmínku je také backendová inicializační funkce `Init`. Tato po konfiguraci a připojení pluginu automaticky nastaví dle aktuálních dostupných profilů databázové tabulky pro data a systémovou tabulku pluginu uvede do výchozího nastavení. V případě, že by se tuto funkci z jakéhokoliv důvodu nepodařilo provést, pak nebude plugin v systému aktivován.

6.2.3 Databáze pluginu

Perzistentní vrstvu pluginu tvoří databázové tabulky v databázi SQLite. Konkrétně se jedná o jednu tabulku udržující systémová nastavení a dále o tabulky udržující data potřebná data pro každý z vytvořených profilů z programu NfSen.

Systémová tabulka nese pojmenování `plugin_system_table`. Tato tabulka je co do počtu sloupců, bohužel, poměrně rozsáhlá. Musí udržovat informace pro každou z kombinací *Flow*, *Packet*, *Traffic* a časového období pro jejich sledování (den, týden, měsíc). V první řadě musí tabulka držet důležitá data o velikosti všech sledovaných odchylek od průměru a jejich nastavených jednotkách. Dále jsou zde tedy uložena data pro zapínání a vypínání viditelnosti jednotlivých křivek v grafech v pluginu. Tato tabulka je při při prvním spuštění pluginu vytvořena a inicializována na výchozí hodnoty funkcí `Init` v backendu pluginu.

Jádrem perzistentní části aplikace jsou však databázové tabulky pro jednotlivé kanály. Pro každý z kanálů je kvůli přehlednosti vytvořena trojice tabulek dle časového období sledování dat. Tedy pro základní profil NfSenu `live` je vytvořena trojice tabulek pojmenovaných `live_average_day`,`live_average_week` a `live_average_month`. Každá z těchto tabulek obsahuje sloupce, které umožňují uložit informace o průměrné velikosti hodnoty `flow`, `packet`, `traffic` v časovém okamžiku `timestamp`, který je zde uložen pro lepší strojové zpracování ve formátu unixové časové značky.

6.3 Návrh pluginu pro vážený průměr

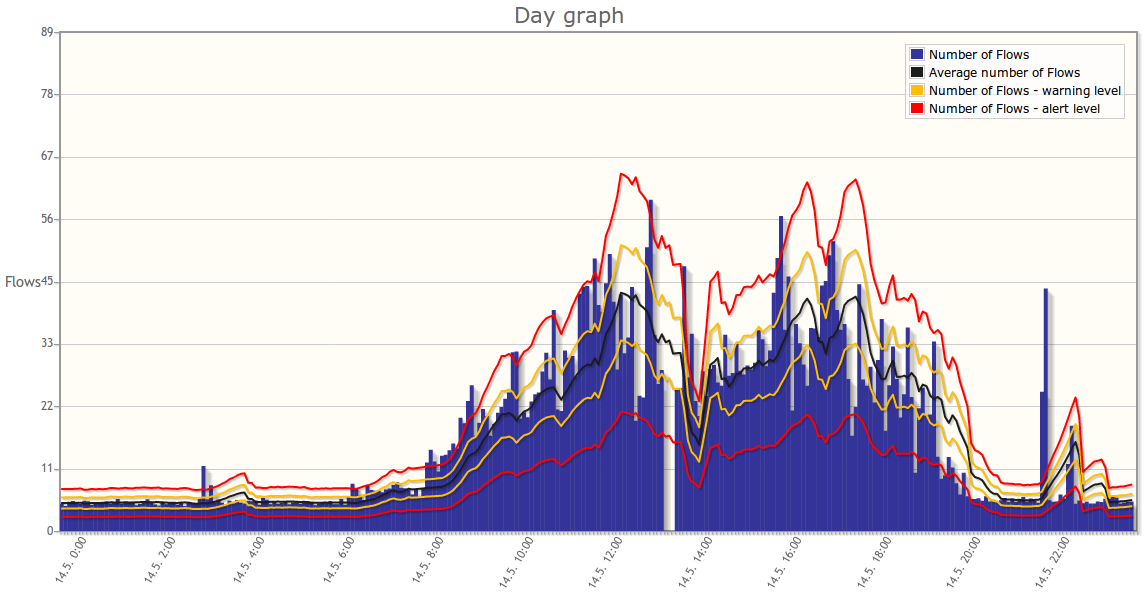
Plugin pro výpočet váženého průměru je pojat poněkud odlišným způsobem než plugin předchozí. Měl by sloužit převážně pro testování vlivu kombinace různě nastavených prahů a přístupů měření na výslednou hodnotu pluginem zjištěné průměrné hodnoty kontrolované veličiny. Proto také obsahuje více nastavení než předchozí plugin zaměřený více na jednoduché ovládání a praktické nasazení v reálném síťovém provozu. Plugin poskytuje veškerou funkcionalitu svého předchůdce a dále ji rozšiřuje v těchto bodech:

- Nastavení vah jednotlivých prvků váženého průměru v normovaném tvaru 4.2.
- Možnost nastavení počtu použitých prvků váženého průměru v rozmezí podporovaném jednotlivými časovými okny.
- Možnost zvolit, zda vybírat prvky na periodické bázi, či použít prosté předchůdce časového okamžiku, pro nějž je hodnota počítána.

6.4 Implementace pluginu pro vážený průměr

6.4.1 Frontend pluginu

Grafický výstup tohoto pluginu je již z povahy věci velmi podobný předchozímu pluginu. Znovu je implementována trojice záložek představujících veličiny sledované pluginem. V každé z těchto záložek se zobrazuje trojice grafů s aktuálními hodnotami, vypočtenou hodnotou průměru a nastavenými odchylkami. Také je u každého grafu zobrazeno informativní okno oznamující kolik prvků je používáno pro výpočet aktuální hodnoty průměru a zda jsou vybírány jako prostí předchůdci, nebo je jejich výběr proveden na dříve popsané periodické bázi.



Obrázek 6.3: Vážený průměr s nastavením 10 předchůdců a rovnoměrně rostoucích vah.

Změny se také udály v záložce *Settings*, kde přibyla další oblast. Zde se nastavuje velikost jednotlivých vah pro výpočet průměru pro denní, týdenní a měsíční časové okno. Také se zde nastavuje způsob výběru prvků pro každé z časových oken.

Provázání skrze časová okna bylo zvoleno proto, aby byla zajištěna nezbytná vnitřní integrita získávaných hodnot. Uživateli je tak v každém okamžiku umožněno porovnávání hodnot přes všechny tři pluginem pozorované hodnoty. Lze tak zjišťovat případné korelace, nebo upravit detektor poplachů tak, aby reagoval ne na konkrétní hodnotu, ale na případné závislosti mezi nimi. Tento plugin také umožňuje nastavení vyvolávání poplachů skrze standardizované rozhraní programu NfSen v záložce *Alerts*.

6.4.2 Backend pluginu

První zásadní odlišností proti pluginu pro výpočet klouzavého průměru je změna vlastní funkce provádějící výpočet průměrné hodnoty (*averageData*). Dalšími podstatnými změnami je změna celková změna periodické funkce `run` tak, aby umožňovala práci s oběma způsoby vybírání prvků pro výpočet průměru a úprava funkce pro výpočet prvků vyhledávaných v RRD databázi programu NfSen dle nastaveného vybíráni prvků.

Také bylo nutné pozměnit implementaci funkce`averageSettingsSave` a také funkce `averageSettingsLoad` tak, aby mohly ukládat (a nahrávat) do systémové tabulky v databázi další hodnoty popisující způsob výběru prvků z databáze a samotné hodnoty vah pro výpočet. Funkce `Init` je upravena a základní nastavení je provedeno tak, že se shoduje s nastavením předchozího pluginu. Je tak jednoduše možné ověřit, že oba pluginy fungují korektně, pokud dodávají shodná data do svých databází.

6.4.3 Databáze pluginu

Databáze opět uchovává tabulku `plugin_system_table`, která obsahuje dodatečné sloupce nesoucí informaci použitém způsobu výběru prvků pro výpočet. Dále obsahuje hodnoty vah jednotlivých elementů, ze kterých jak už bylo řečeno je také určován počet použitých elementů. Dále jsou přítomny tabulky pro trojici časových oken každého zpracovávaného kanálu se sloupci pro průměrnou velikost toků, počtu paketů a počtu Bytů. Každý uložený řádek je opatřen časovým údajem ve formátu unixové časové značky.

6.5 Návrh pluginu pro metodu ASTUTE

Tento plugin je navržen pro ověření chování metody ASTUTE a jejich detekčních schopností v počítačové síti. V případě prokázání jejich detekčních schopností, pak může být použit v produkčním prostředí správci počítačové sítě pro detekci anomálií.

Plugin spolupracuje se standardním profilem *Live*. Je možné plugin použít i na další z profilů. Jen je třeba mít na paměti, že plugin potřebuje mít přístup k neagregovaným datům. Předem agregovaná data totiž zkreslí výsledky výpočtů metodou ASTUTE. Mezi funkcionalitu poskytovanou pluginem patří:

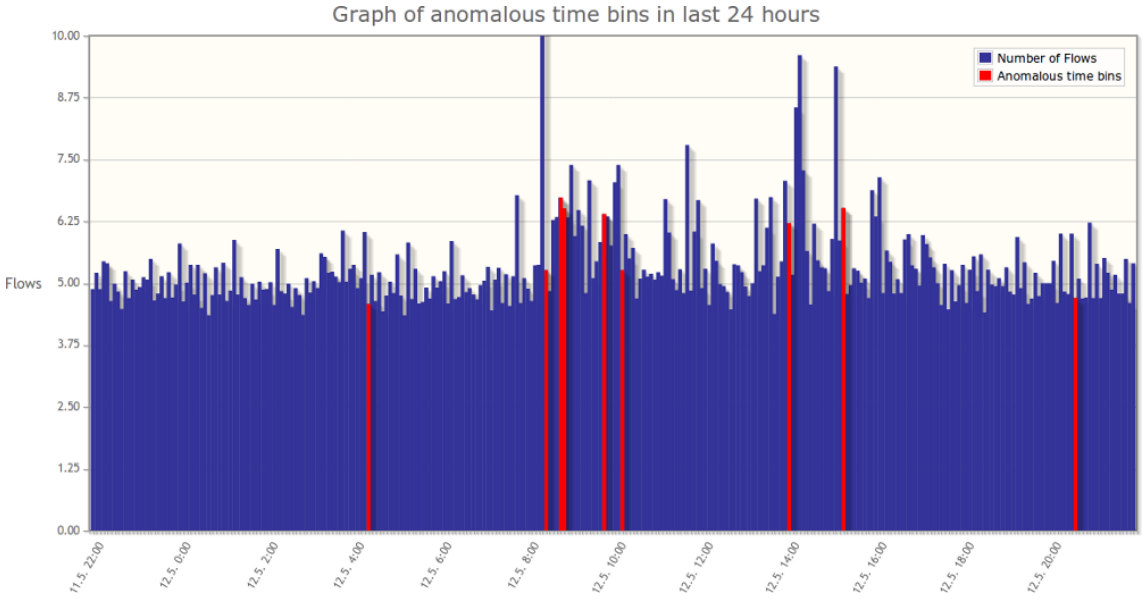
- Poskytuje grafický přehled všech detekovaných anomálních časových oken za posledních 24 hodin.
- Umožňuje nechat si vypsat anomální časová okna s podrobnými informacemi o detekované události v rámci okna.
- Umožňuje nastavit odeslání upozornění emailem, či provést jinou akci při detekci anomálního časového okna.
- Dovoluje nastavení hodnoty $K(p)$ a agregačních parametrů uživatelem.
- Umožňuje uživateli vymazat databázi měření.
- Databáze ukládá všechny naměřené hodnoty ASTUTE assessment value (dále AAV) a umožňuje jejich zpětnou analýzu.

6.6 Implementace pluginu pro metodu ASTUTE

6.6.1 Frontend pluginu

Hlavní záložkou v pluginu je *Overview*. Obsahuje graf vykreslující toky za posledních 24 hodin. V tomto grafu jsou pomocí sloupcového grafu červené barvy zachycena detekovaná anomální časová okna. Pod tímto grafem jsou pak vykresleny grafy zobrazující hodnoty AAV při všech použitých úrovních agregace na všech dostupných kanálech profilu. Tabulka

na konci záložky pak umožňuje vypsání detailních informací o detekovaných anomálních časových oknech (kanál profilu, hodnota prahu $K(p)$, agregace, která detekovala anomálii, výpis všech naměřených AAV hodnot).



Obrázek 6.4: Zobrazení anomálních časových okamžiků odhalených pomocí ASTUTE.

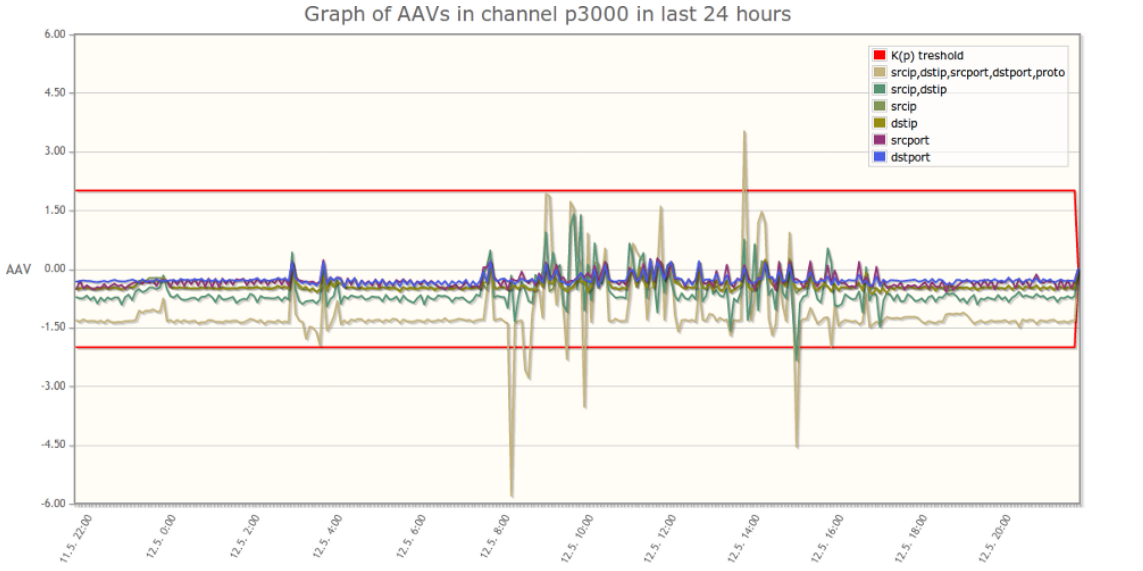


Table of the last 5 anomalous time bins

I would like to display last 5 anomalous time bins. Submit

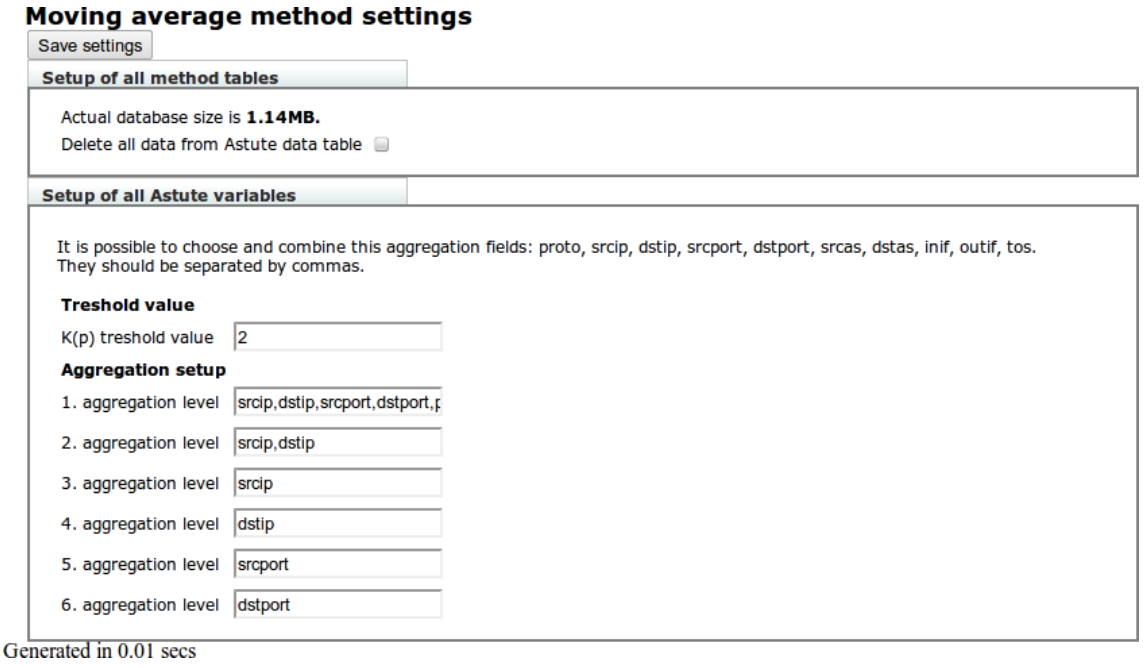
Time of discovery	Channel	Discovered by	K(p)	Aggregation 1	Aggregation 2	Aggregation 3	Aggregation 4	Aggregation 5	Aggregation 6
2012-05-12T20:15:00+02:00	p3001	srcip,dstip,srcport,dstport,proto	2.000	-2.054	0.279	-0.356	-0.416	0.213	0.213
2012-05-12T14:55:00+02:00	p3000	srcip,dstip,srcport,dstport,proto srcip,dstip	2.000	-4.546	-2.338	-0.617	-0.502	-0.259	-0.347
2012-05-12T13:40:00+02:00	p3000	srcip,dstip,srcport,dstport,proto	2.000	3.518	0.748	-0.084	0.022	0.095	0.027
2012-05-12T09:50:00+02:00	p3000	srcip,dstip,srcport,dstport,proto	2.000	-3.526	-1.061	-0.362	-0.278	-0.217	-0.277
2012-05-12T09:25:00+02:00	p3000	srcip,dstip,srcport,dstport,proto	2.000	-2.308	-1.093	-0.485	-0.576	-0.374	-0.404

Obrázek 6.5: Zobrazení hodnot AAV a tabulka s výpisem anomálních časových okamžiků.

Druhá ze záložek, *Settings*, zobrazuje v první oblasti aktuální velikost databáze pluginu a umožňuje mazání tabulky s naměřenými hodnotami. Druhá oblast obsahuje nastavení proměnných pro výpočet metody ASTUTE. Mezi nejdůležitější nastavitelné parametry patří práh $K(p)$. Tento práh řídí rozhodování o anomálnosti časového okna dle v něm naměřených hodnot. Dále je možné nastavit libovolnou kombinaci položek 6.1, dle kterých bude prováděna agregace dat pro výpočet metody ASTUTE. Případně je možné nechat pole prázdné a snížit tak počet použitých úrovní agregace z autory doporučovaných šesti.

Položka agregace	**Význam položky v programu Nfdump**
proto	Protokol detekovaný v síťovém toku (TCP, UDP, ICMP, atd.).
srcip	Zdrojová adresa síťového toku ve formátu IPv4/IPv6.
dstip	Cílová adresa síťového toku ve formátu IPv4/IPv6.
srcport	Zdrojový port (0-65535) v síťovém toku.
dstport	Cílový port (0-65535) v síťovém toku.
srcas	Zdrojové AS číslo.
dstas	Cílové AS číslo.
inif	Číslo vstupního SNMP rozhraní.
outif	Číslo výstupního SNMP rozhraní.
tos	Typ služby. Platí pro TCP v IPv4.

Tabulka 6.1: Položky použitelné pro agregaci toků při výpočtu metody ASTUTE.



Obrázek 6.6: Záložka *Settings* s nastaveními pro metodu ASTUTE.

6.6.2 Backend pluginu

V backendové části pluginu jsou použity dvě uživatelem volané funkce pro ukládání a načítání nastavení z databáze. Další dvě uživatelské funkce jsou volány za účelem vykreslování

grafů (`getAstuteData`, `getAstuteAAVs`) a vypisování tabulky anomálních časových toků (`getAstuteTable`).

Periodicky volanou funkcí je pak funkce `alert_condition` kontrolující v databázi existenci záznamu o anomálnosti pro aktuální časový okamžik a umožňující nastavování poplachů standardní cestou v uživatelském rozhraní NfSenu. Periodická funkce `astute_run` zpracovává toky dle algoritmu 4.2.2, kde v závěru porovnává vypočtené hodnoty 4.7 s uživatelem nastavenou hodnotou $K(p)$. Celý výpočet je pak proveden na všech nastavených úrovních agregace a dostupných profilech dle algoritmu 6.6.2.

Algorithm 3 Periodický výpočet hodnot metody ASTUTE.

Vstup: název NfSen profilu, název skupiny profilů
Ověř existenci potřebných tabulek v databázi.
Získej informace o kanálech patřících do profilu.
Načti z tabulky *plugin_system_table* hodnotu $K(p)$ a platné úrovně agregace.
for all kanály ze zpracovávaného profilu **do**
  for all platné úrovně agregace **do**
    Získej toky vyhovující agregaci z aktuálního časového okna (i).
    Získej toky vyhovující agregaci z časového okna $i-1$ až i.
    Převeď toky obou případů do samostatných *piddle*.
    Proveď výpočet těchto *piddle* dle algoritmu detektoru ASTUTE (4.2.2).
  end for
end for
Ulož časovou značku, vypočtené AAV, rozhodnutí o anomálnosti časového okna, případně agregační úroveň a kanál, kde byla objevena anomálie do databáze.
Aktualizuj velikost databáze v tabulce *plugin_system_table*.

6.6.3 Databáze pluginu

Databázová část pluginu implementujícího metodu ASTUTE se skládá opět z tabulky `plugin_system_table`, která uchovává v poli `kp` informace aktuálně platné hodnotě prahu $K(p)$, potřebného pro zjištění zda se jedná o anomálii. Dále se pak v polích `aggregation_0` až `aggregation_5` uchovávají agregační pole programu Nfdump pro všech šest úrovní agregace. Ke každému řádku je pak přiložena velikost celé databáze a časová značka označující počátek platnosti tohoto nastavení ve formátu unixové časové značky.

Druhá tabulka `live_astute` obsahuje vlastní data vytvořená metodou ASTUTE při sledování počítačové sítě. Každý řádek je opatřen ve sloupci `timestamp` časovou značkou ve formátu unix timestamp, označující korespondující časový slot v databázi metody ASTUTE se souborem v adresářové struktuře dat programu Nfdump. Také je uložena informace, ve kterém kanálu profilu byla hodnota naměřena `channel`, zda se jedná o anomální časový slot `anomalous` (hodnota True/False), který ze stupňů agregace jej takto označil `discovered_by` a jaká hodnota prahu `kp` byla pro rozhodnutí použita. Dále jsou uloženy všechny hodnoty AAV pro příslušnou použitou agregaci.

Při prvním startu pluginu je pak vytvořena tabulka pro vypočtená data. Jsou také nastaveny základní hodnoty potřebné k výpočtu dle dříve diskutované teorie metody ASTUTE (šestice agregačních stupňů [24] a hodnota $K(p) = 2$ [24]).

Kapitola 7

Testování detekčních metod

V následující kapitole jsou popsány testy provedené na implementovaných metodách za účelem odhalení jejich detekčních schopností. První část se zaměřuje na rozbor dat získaných metodou ASTUTE a dále je demonstrován vliv zvolených váhových koeficientů na schopnost detekce pomocí váženého průměru. Druhá část kapitoly se zaobírá provedenými pokusnými útoky na síť monitorovanou pomocí implementovaných metod a vyhodnocení jejich schopnosti tyto útoky detekovat.

7.1 Rozbor dat získaných metodou ASTUTE

Následující dvě analýzy jsou provedeny na datech získaných pomocí pluginu implementujícího metodu ASTUTE v síti společnosti INVEA-TECH a.s. Poslední měření a analýza pak byla provedena na poskytnutých vzorcích anonymizovaných dat, získaných ze sondy umístěné v síti na Fakultě informačních technologií na Vysokém učení technickém v Brně.

7.1.1 Rozbor počtu odhalených anomálních časových oken

První analýza proběhla na datech z kanálů p3000 a p3001 v profilu *live* v období 12.dubna 2012 (20:05) až 7.května 2012 (09:25). Jejím smyslem bylo získat přesný přehled o podílu jednotlivých stupňů agregace na detekci anomálních časových oken zjištěných touto metodou. Výsledkem analýzy pak je tabulka číslo 7.1 zobrazující procentuální rozložení detekovaných anomálních časových oken vzhledem k použitému stupni agregace v průběhu výpočtu.

Jak již bylo dříve řečeno, o anomálii se jedná, pokud vypočtená hodnota AAV při nastavené agregaci překročí prahovou hodnotu $K(p)$, která byla nastavena na doporučenou velikost 2.

Agregace	**Kanál p3000**	**Kanál p3001**
srcip, dstip, srcport, dstport, proto	78.57%	100.00%
srcip, dstip	20.72%	0.00%
srcip	0.00%	0.00%
dstip	0.00%	0.00%
srcport	0.29%	0.00%
dstport	0.42%	0.00%

Tabulka 7.1: Detekce anomálních časových oken pomocí nastavených agregací.

Výsledné zjištění poukazuje na jasnou převahu anomálií detekovaných pomocí agregace `srcip, dstip, srcport, dstport, proto` v obou kanálech, což je dle dalších pozorování také způsobeno charakterem provozu v sítí, který se z velké části sestává z HTTP a HTTPS komunikace. Tato je uskupena do sessions. Následkem toho jevu dochází k porušení Teorému 1 a generování falešných poplachů.

7.1.2 Vliv hodnoty prahu K(p) na počet zjištěných anomálií

Při druhé analýze provedené také na datech z kanálů p3000 a p3001 profilu *live* v období 12.dubna 2012 (20:05) až 7.května 2012 (09:25) byl zjišťován vliv nastavení prahu $K(p)$ na počet detekovaných anomálních časových oken a statistické veličiny popisující soubor získaných dat.

Nejprve jsou představeny tabulky obsahující významné veličiny popisující data získaná z obou sledovaných kanálů. Konkrétně se jedná se o průměr ($\bar{x}$), medián ($\tilde{x}$), minimum $X_{(1)}$, maximum $X_{(n)}$ a směrodatnou odchylku (σ) naměřených hodnot. Tato čísla poskytují nejen důležité shrnutí informací o provozu v monitorovaných kanálech p3000 a p3001, ale jsou podkladem pro následují část prováděné analýzy.

Agregace	$\bar{x}$	$\tilde{x}$	σ	$X_{(1)}$	$X_{(n)}$
srcip, dstip, srcport, dstport, proto	2.163	1.691	1.131	0.001	30.298
srcip, dstip	0.962	0.802	0.468	0.000	6.609
srcip	0.412	0.450	0.126	0.001	1.035
dstip	0.428	0.474	0.122	0.000	0.914
srcport	0.329	0.314	0.103	0.000	2.401
dstport	0.324	0.313	0.086	0.000	2.510

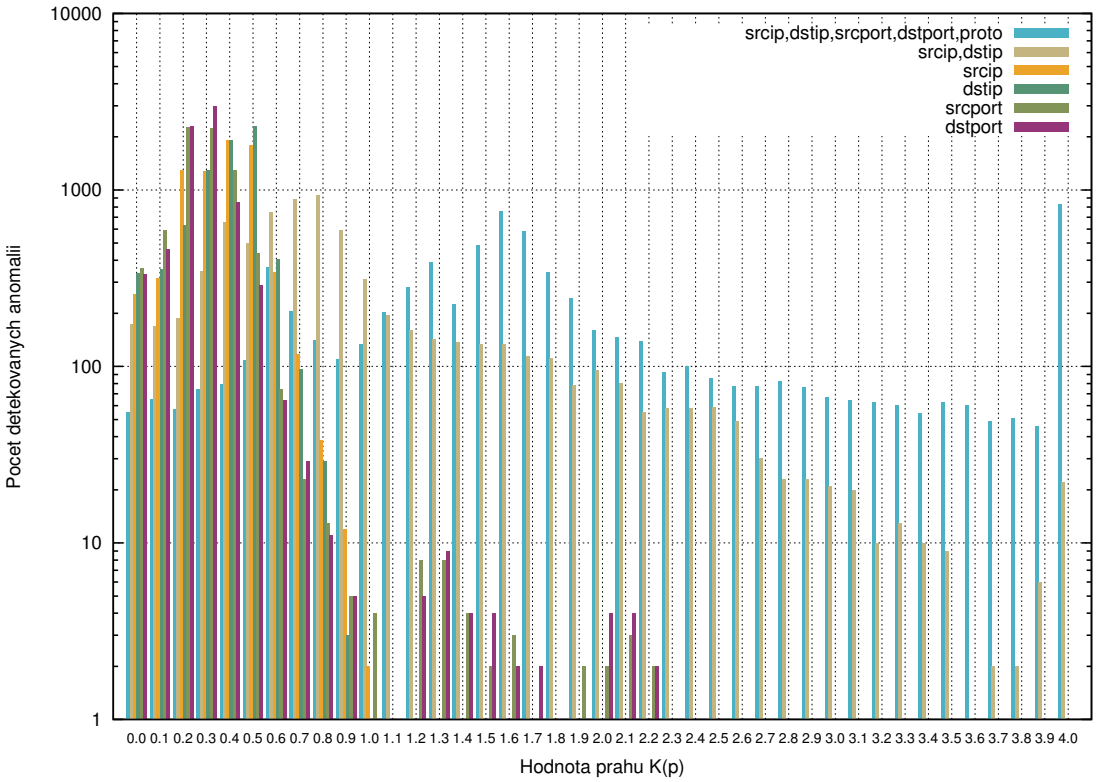
Tabulka 7.2: Statistický popis anomálií v kanálu p3000.

Agregace	$\bar{x}$	$\tilde{x}$	σ	$X_{(1)}$	$X_{(n)}$
srcip, dstip, srcport, dstport, proto	1.637	1.371	0.918	0.000	13.690
srcip, dstip	0.429	0.227	0.180	0.001	1.701
srcip	0.355	0.119	0.090	0.000	0.759
dstip	0.347	0.120	0.092	0.000	0.698
srcport	0.226	0.082	0.060	0.000	1.466
dstport	0.243	0.094	0.071	0.000	1.386

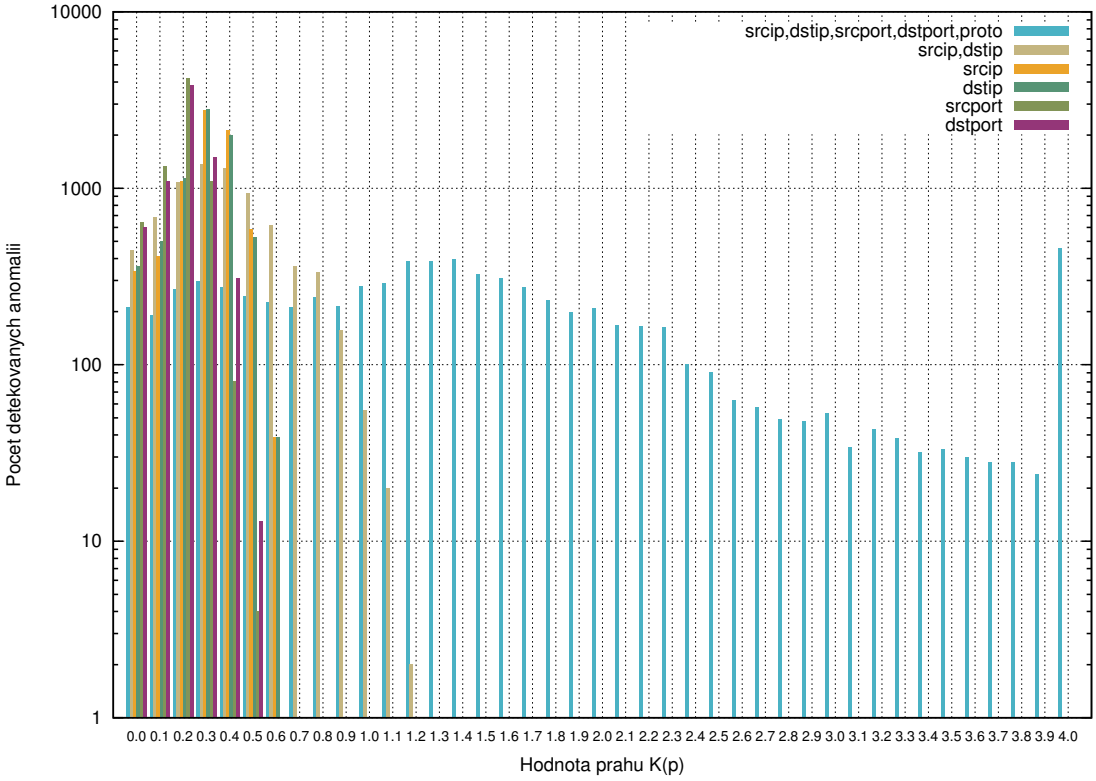
Tabulka 7.3: Statistický popis anomálií v kanálu p3001.

Další část analýzy představuje histogramy zachycující počet anomálií detekovaných v kanálech v závislosti na velikosti prahu $K(p)$. Hodnoty AAV získané z jednotlivý úrovní agregace jsou použity ve své absolutní hodnotě ve shodě s postupem detekce v algoritmu 4.2.2. Jako šířka intervalu byla zvolena hodnota $0,1$ a počet prvků 42. Takto zvolené koeficienty umožňují zachytit v histogramu přesně rozsah hodnot v rozmezí $\bar{x} - 2\sigma$ až $\bar{x} + 2\sigma$. Jedná se cca o $90 - 95\%$ všech hodnot a odpovídá tak empirickému pravidlu 2σ.

Z měření je patrné, že zvolená hodnota detekčního prahu poskytuje dostatečné množství anomálií a je tedy možné ji ponechat na hodnotě doporučované autory článku. Poměrně zajímavým jevem je pak naprosto odlišná charakteristika prvního agregačního stupně `srcip, dstip, srcport, dstport, proto` od ostatních stupňů. Dle provedených pozorování je



Obrázek 7.1: Histogram detekovaných anomálií pro kanál p3000.



Obrázek 7.2: Histogram detekovaných anomálií pro kanál p3001.

toto způsobeno falešnými poplachy vyvolanými procházením webových stránek v důsledku malého provozu v síti a tedy relativně malého vzorku dat. Pro podporu tohoto tvrzení je možné uvést, že se zvýšený počet poplachů vyskytoval pouze v pracovní době organizace. Ve dnech pracovního klidu a v období 18:00-08:00 bylo jejich pozorované množství minimální.

7.1.3 Výsledky měření ze silného provozu

Úkolem tohoto měření provedeného na anymizovaných datech získaných z provozu FIT VUT v Brně je zjištění chování metody ASTUTE v síti se silnějším provozem, než byla standardně po celou dobu tvorby této práce nasazena. Dalším důvodem bylo ověření hypotézy, že za počet falešných poplachů může právě charakteristika provozu v síti INVEA-TECH, jež se sestává v drtivé většině z komunikace webových prohlížečů. Ostatní metody v tomto provozu nebyly testovány. Z jejich podstaty na ně nemůže mít změna provozu takový dopad jako metodu ASTUTE. V jejich případě by se pouze jednalo o úpravu prahů v záložce *Settings*.

Agregace	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
flow*	-1.757	-1.975	-1.465	-1.338	-1.587	-1.618	-1.102	-1.465	-1.530	-1.889
srcip,dstip	-1.136	-0.846	-0.743	-0.844	-0.911	-0.802	-0.660	-0.743	-0.677	-0.661
srcip	-1.418	-1.225	-1.170	-1.110	-1.137	-1.182	-1.125	-1.061	-0.887	-0.896
dstip	-1.174	-1.048	-0.953	-0.909	-0.999	-0.927	-0.664	-0.934	-1.284	-0.776
srcport	-1.132	-1.339	-1.250	-1.440	-1.864	-1.601	-1.270	-1.523	-1.476	-1.434
dstport	-1.447	-1.028	-1.518	-1.799	-1.081	-1.617	-1.154	-1.416	-1.676	-1.471

** srcip, dstip, srcport, dstport, proto*

Tabulka 7.4: Hodnoty AAV naměřené v datech z VUT FIT.

Z provedených měření vyplývá, že provoz na síti se silný provozem jako je ten na FIT VUT v Brně vykazuje poněkud odlišné hodnoty AAV než provoz v malé síti, kde proběhla většina měření, přesto však na dodaných vzorcích nebyla pozorována výraznější odchylka od dříve pozorovaných hodnot. Na tomto dodaném vzorku dat také nebyly metodou ASTUTE nalezeny žádné anomálie. Zcela přesné vyhodnocení funkčnosti by však na této síti bylo možné dát až po jejím podrobném monitorování.

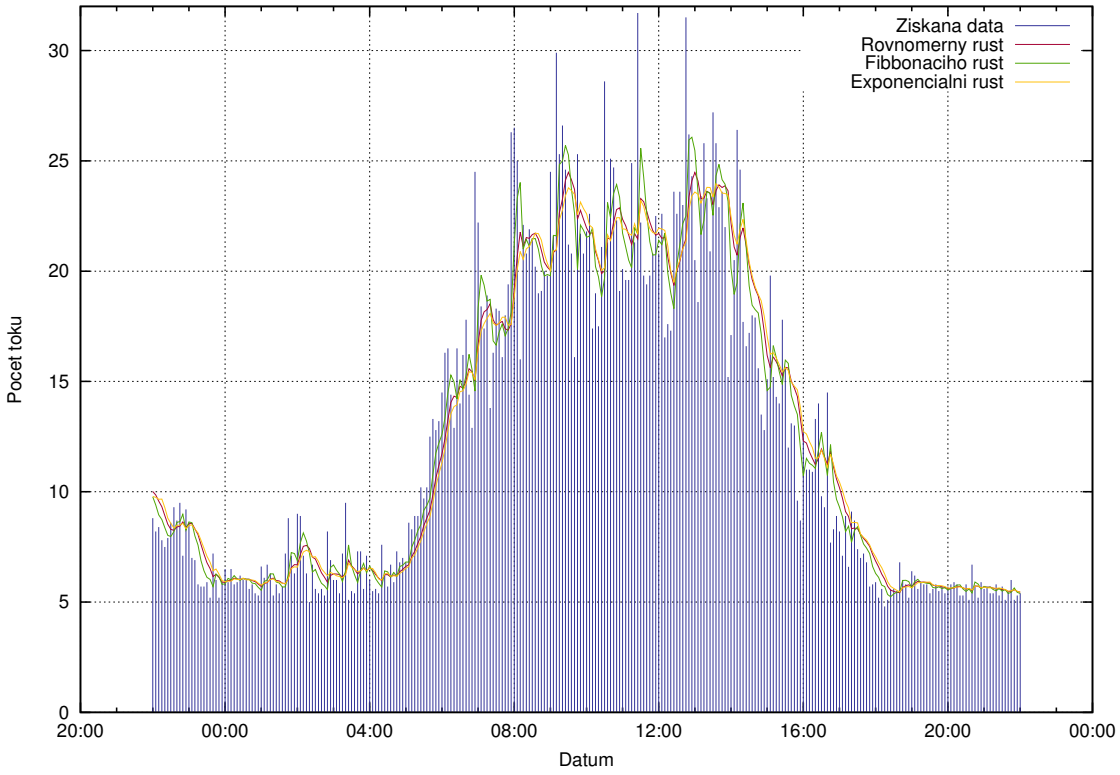
7.2 Vliv váhových koeficientů průměru na detekci anomálií

V tomto analýze byl porovnáván vliv rozdílných sad vah patřících k jednotlivým prvkům na výpočet váženého průměru. Měření proběhlo na typickém zástupci získaných dat (26.4.2012 00:00-24:00). Pro samotný test byly zvoleny následující parametry. Prvky pro výpočet průměrné hodnoty byly voleny jako prostí předchůdci hledané hodnoty. Byl určen shodný počet 10 prvků pro všechna časová okna. Dále pak byly vybrány normované váhy, jež tvoří tři výrazné posloupnosti. První je posloupnost s rovnoměrným přírůstkem. Následují váhy tvořící Fibbonacciho posloupnost. Jako třetí byly použity váhy s posloupností odpovídající exponenciálnímu růstu. Hodnoty všech posloupností jsou pak shrnuty v níže uvedené tabulce 7.5.

Z těchto tří testovaných posloupností vykazovaly nejzajímavější vlastnosti první a poslední z nich. Při posloupnosti představující rovnoměrný růst byla data sledována konzervativně a byly hlášeny veškeré významnější odchylky od normálu po celou dobu jejich trvání. Naopak posloupnost představující exponenciální růst se těmto odchylkám přizpůsobovala. Detekovala vždy pouze nástupní hranu a další jejich průběh byl již ignorován.

Název posloupnosti	Přesný formát posloupnosti
Rovnoměrný růst	0,019; 0,036; 0,055; 0,07; 0,09; 0,11; 0,13; 0,15; 0,16; 0,18
Fibbonacciho růst	0,006; 0,006; 0,013; 0,02; 0,034; 0,055; 0,09; 0,156; 0,237; 0,383
Exponenciální růst	0,061; 0,068; 0,075; 0,083; 0,091; 0,1; 0,112; 0,123; 0,136; 0,151

Tabulka 7.5: Hodnoty vah představených posloupností pro výpočet s 10 prvky.

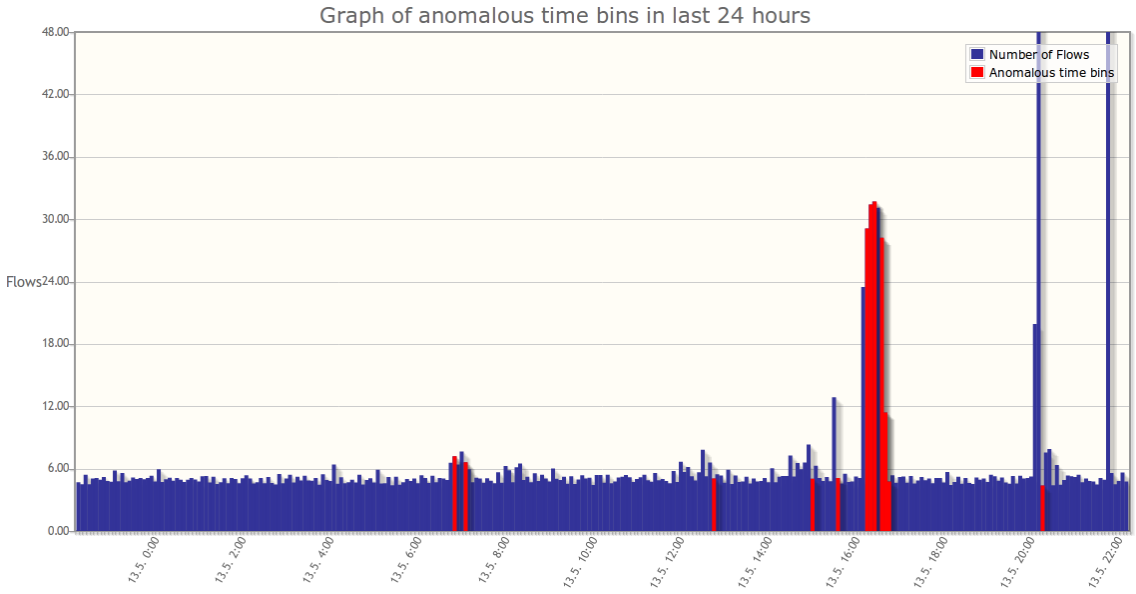


Obrázek 7.3: Předpověď počtu toků s použitím vah z předchozí tabulky.

7.3 Detekce skenování

Úkolem testu je porovnání schopnosti implementovaných metod odhalit skenování strojů z vnějšku monitorované počítačové sítě. Pro provedení byl použit obecně známý nástroj *Nmap*. Test proběhl jako série tří po sobě následujících skenování jednoho stroje simulujících útočníka nejprve hledajícího otevřené porty a poté zjišťujícího verze služeb běžících na těchto portech. Tento test byl třikrát opakován. Na přiloženém obrázku 7.4 z prostředí pluginu jsou zachyceny první dva pokusy (13.5. 20:00-20:15 a 13.5. 21:50-22:00). Třetí vykazoval také obdobnou charakteristiku. Skenování bylo provedeno v následujícím pořádí a konfiguraci skenů:

1. `nmap -sS -O att.acked.com`
2. `nmap -sSU -O att.acked.com`
3. `nmap -sV -p 22,53,80,443,110,143,2200,4564 att.acked.com`



Obrázek 7.4: Monitoring skenování portů v pluginu ASTUTE (pravý okraj grafu).

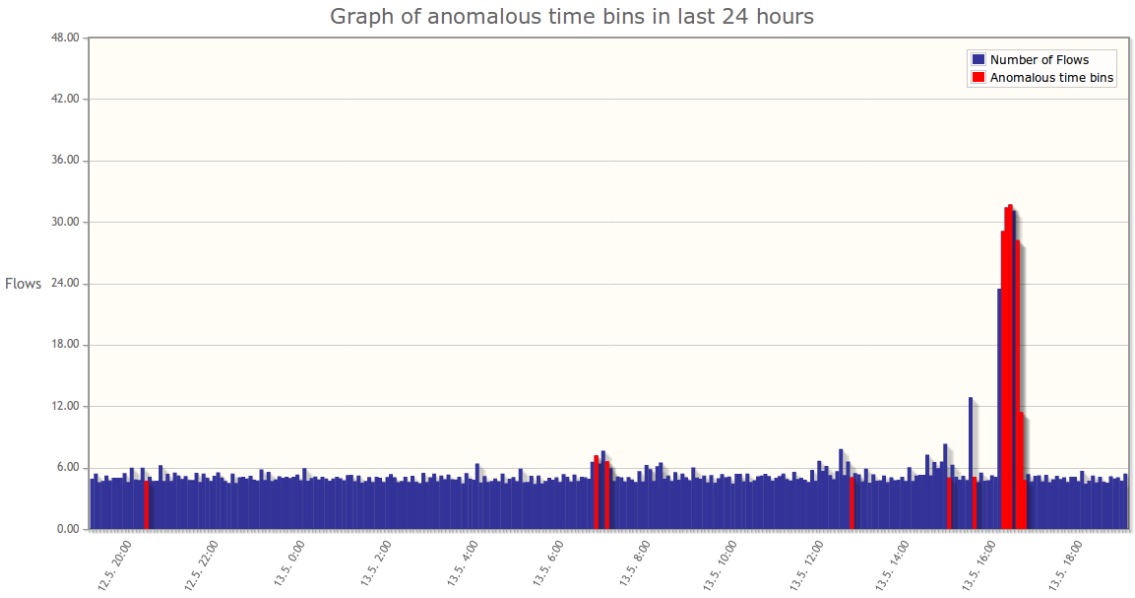
Dvě modré špičky v pravé části grafu značí skeny číslo 1 a 2. dále lze jasně vyčíst, že metoda ASTUTE detekovala pouze poslední ze série skenů a to pouze v prvním případě. Tento výsledek se již bohužel nepodařilo v dalších testech replikvoat. Proto je úspěšnost této metody sporná. Pluginy spoléhající na aplikaci jednodušších metod odhalily velmi dobře první ze série skenů díky jeho velkému počtu toků. Ostatní zůstaly nepovšimnuty. Také v ostatních sledovaných veličinách nebyly zaznamenány žádné anomálie způsobené skenováním stroje.

7.4 Detekce HTTP flood útoku

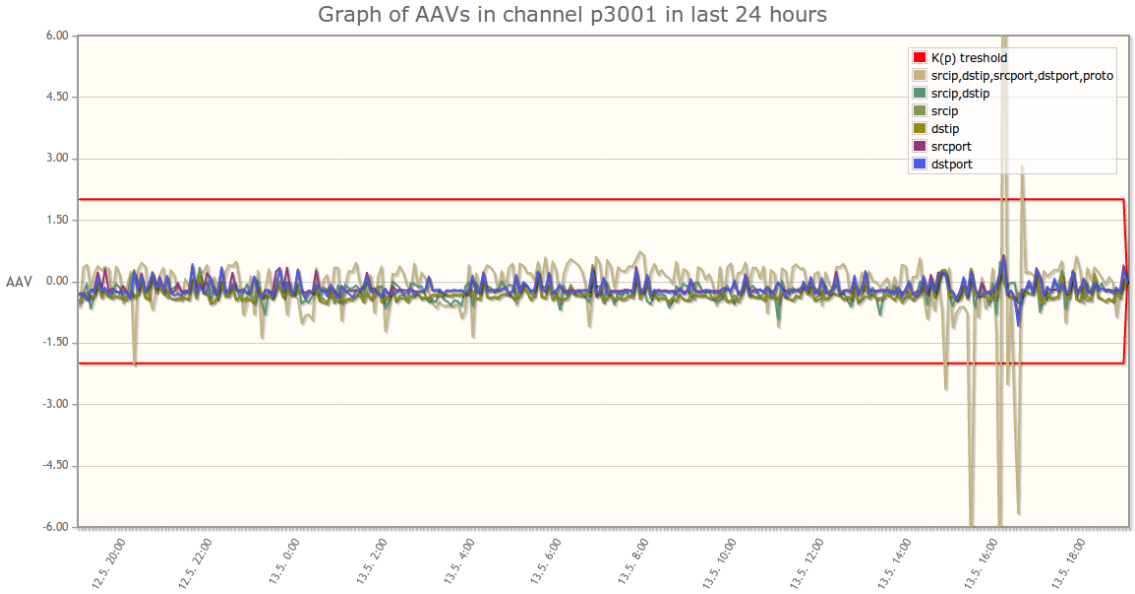
Tento test se zaměřil na srovnání všech tří pluginů vzhledem k detekci HTTP flood útoku. Pro útok byl použit běžně dostupný nástroj *Slowloris* [23], který útočí pomocí zasílání nekompletních HTTP požadavků a snaží se tak o vyčerpání zdrojů cílového stroje. Jeho výhodou je odesílání poměrně nízkého množství dat, což se také potvrdilo v průběhu vlastního testu.

Z porovnání všech tří metod vychází velmi dobře metoda ASTUTE, která dokázala sledovat útok po celou dobu jeho trvání a přesně jej označit. Detekovat se jej podařilo díky agregační úrovni `srcip, dstip, srcport, dstport, proto`. Naměřené hodnoty AAV byly zvláště v první části útoku natolik vysoké, že by jej spolehlivě odlišily od případného falešného poplachu 7.6.

Detekční metoda používající pohyblivý průměr nedetekovala útok v rámci žádné z měřených veličin. V počtu toků byl útok zamaskován, protože nedosáhl hodnoty průměru toků v předchozích dnech. V ostatních dvou sledovaných veličinách (počet paketů, počet Bytů) se útok vůbec neprojevil. Naopak metoda používající vážený průměr útok detekovala pomocí vhodně nastavených parametrů (10 prvků, předcházející prvky a rovnoměrný růst vah) v počtu toků. Zbylé dvě sledované veličiny opět nenaznačily nic neobvyklého.



Obrázek 7.5: Zobrazení HTTP flood útoku v prostředí pluginu ASTUTE.



Obrázek 7.6: Hodnoty AAV v kanálu p3001 v průběhu útoku.

Detekováno	Kanál	|AAV|
16:35:00	p3000	2.800
16:35:00	p3001	2.842
16:30:00	p3000	5.175
16:30:00	p3001	5.669
16:25:00	p3000	2.921
16:25:00	p3001	3.297
16:15:00	p3000	2.440
16:15:00	p3001	2.512
16:10:00	p3000	13.175
16:10:00	p3001	15.007
16:05:00	p3001	8.188

Tabulka 7.6: Přehled hodnot AAV v průběhu detekce útoku.

7.5 Detekce slovníkového ssh útoku

Smyslem tohoto testu je snaha zjistit, zda zvládnou implementované detekční metody odhalit probíhající snahu útočníka o uhodnutí hesla pro přístup ke službě ssh pomocí slovníkového útoku. K provedení testu byl zvolen poměrně nový nástroj *Ncrack* od autorů v této práci dříve použitého skeneru *Nmap*. Zásadní výhodou tohoto programu je, že umožňuje rozložit útok do delšího časového období a také nastavovat jeho charakteristiky, což podstatně zvyšuje možnost vyhnutí se odhalení detekčním systémem.

Samotný útok byl realizován ve ve dvou časových obdobích (21:25-21:30, 22:00-21:05) s nastaveným přepínačem `-T2` a poté ve dvou časových obdobích s přepínačem `-T1` (22:55-23:00, 23:05-23:45). Tento parametr představuje dříve zmíněnou časovou kontrolu útoku a nabývá hodnot 0 až 5, kde 3 pak představuje normální nastavení. Autoři uvádějí, že při nastavení hodnoty na 0 a 1 je možné se vyhnout detekčním mechanizmům v nasazených IDS systémech [12].

Čas detekce	Kanál	|AAV|
23:45:00	p3001	2.512
23:40:00	p3000	2.463
23:40:00	p3001	3.039
23:35:00	p3000	2.865
23:35:00	p3001	3.528
23:00:00	p3001	2.217
22:55:00	p3001	2.103

Tabulka 7.7: Hodnoty AAV v průběhu detekce SSH útoku s parametrem `-T1`.

Metodě ASTUTE se podařilo tyto útoky detekovat ve všech případech v celé jejich délce, přestože se neprojevovaly významným způsobem v žádné ze sledovaných tří sledovaných veličin. Proto také útoky nebyly odhaleny ani jednou z metod používajících k detekci pohyblivý (vážený) průměr. Druhá sada útoků s nastaveným parametrem pro vyhnutí se IDS bovšem již nezanechala tak výraznou stopu jako přecházející útok. Hodnoty AAV naměřené pří těchto útocích je možné porovnat v tabulkách 7.7 a 7.8. Všechny tyto útoky byly detekovány díky agregační úrovni `srcip, dstip, srcport, dstport, proto`.

Čas detekce	Kanál	|AAV|
22:05:00	p3000	8.440
22:05:00	p3001	8.901
22:00:00	p3000	6.270
22:00:00	p3001	5.766
21:30:00	p3000	2.699
21:30:00	p3001	10.473
21:25:00	p3001	9.123

Tabulka 7.8: Hodnoty AAV v průběhu detekce SSH útoku s parametrem `-T2`.

7.6 Detekce velkého datového přenosu

Tento test byl zaměřen na schopnost metod detekovat velké datové přenosy. Ve své podstatě nemusí být takový přenos způsoben anomálií či útokem. Přesto může být schopnost detekovat takovou událost shledána užitečnou, proto je pro doplnění celkového obrazu schopností metod tento pokus proveden. Vlastní test byl spuštěn 12.5. ve 23:51 s předpokladem, že se přenos skryje do pravidelného zvýšeného půlnočního provozu v síti způsobeného zálohováním dat. Přenos představovalo stáhnutí instalačního DVD systému Fedora 16 o velikosti 3,5GB s dobou přenosu cca 4 minuty.

Detekční metoda založená na průměru tuto metodu detekovala díky počtu paketů a Bytů, které překročily průměrnou hodnotu zálohování. Metoda využívající vážený průměr přenos detekovala shodně na počtu paketů a Bytů díky porušení průměru získaného z předchozích nízkých hodnot. Při obou metodách se tento přenos v počtu toků nijak výrazněji neprojevil. Implementace metody ASTUTE na tento extrémní datový přenos nereagovala, na což upozorňují i její autoři ve svém článku. Jeden silný tok nemůže porušit předpoklad nezávislosti a tak způsobit porušení metody ASTUTE a způsobit tak své odhalení [24].

Kapitola 8

Možná budoucí rozšíření

Kapitola krátce nastiňuje možná budoucí rozšíření této diplomové práce a celkovou vizi uplatnění těchto a obdobných statistických metod pro monitorování provozu ve vysokorychlostních sítích s hustým provozem jako jsou páteřní sítě, sítě jednotlivých poskytovatelů internetového připojení a také podnikové sítě. Tato možnost se zdá obzvláště příhodná vzhledem k přístupu metod, který zajišťuje monitorování sítí bez porušování zákonů a směrnic o ochraně osobních údajů. Také je třeba připomenout, že tyto metody jsou schopny díky svému založení detekovat *zero day* útoky, které se mohou vyhnout standardním IDS systémům. Mohou tedy být použity k objevování nových a vysoce sofistikovaných způsobů útoku na počítačové sítě, které se zatím úspěšně vyhýbaly odhalení.

8.1 Pluginy pro klouzavý průměr a vážený klouzavý průměr

Zvláště u pluginu používajícího vážený průměr je třeba blíže prozkoumat vliv rozličného množství nastavení pro detekci anomálií v síti. Bylo by vhodné vytvořit několik přednastavených profilů pro sítě různé velikosti a charakteristiky provozu a usnadnit tak běžným uživatelům konfiguraci a vyladění pluginu dle jejich potřeb.

Z implementačního hlediska se zdá zajímavou možností vytvoření pluginu jako frameworku pro aplikaci statistických metod nad nástrojem NfSen, kde by dle potřeby docházelo pouze k výměně mikromodulů obsahujících příslušnou metodu detekce anomálií.

8.2 Plugin pro metodu ASTUTE

Vzhledem k tomu, že metoda prokázala během práce zajímavé detekční schopnosti, pak dalším krokem by mělo být její porovnání s ostatními specifickými metodami jako je Kalmanův filtr a algoritmus Holt-Winters. Pokud by byl vyřešen z pohledu implementace drobný problém s požadavkem metody na práci s toky překračujícími hranici časového okna, který není v programu Nfdump dostatečně dořešen, pak bylo možné dále zvýšit rychlost zpracování velkých objemů dat a metodu poté použít k monitorování sítí s opravdu silným provozem.

Kapitola 9

Závěr

Na základě prostudování vybraných hrozeb v počítačové síti, existujících statistických metod pro zpracování dat, technologie NetFlow a detekční metody ASTUTE, byla vytvořena sada tří pluginů programu NfSen implementujících metody popsané v první části práce.

Dva z těchto pluginů využívají metody běžně používané k analýze finančních a jiných dat. Aplikují je však na poli detekce a monitorování počítačových sítí skrze typické hodnoty zachytávané v NetFlow datech. První, jednodušší, z pluginů využívá metodu váženého průměru a byl vyvíjen v rámci Community programu firmy INVEA-TECH a.s.. Je tak částečně uzpůsoben jejím potřebám a přáním. Druhý, z něj vycházející, plugin implementuje metodu váženého průměru, která umožňuje větší svobodu nastavení a celkově je zaměřena více experimentálním směrem. Dovoluje uživateli testovat nesčetné kombinace možností a je tak potencionálně velmi silným nástrojem, který již ovšem potřebuje hlubší znalosti pro jeho dokonalé zvládnutí. S výhodou lze pro nastavování pluginu použít analogie s burzovními daty, která vykazují v jistých směrech obdobné charakteristiky.

Poslední vytvořených z pluginů využívá pro detekci anomálií odlišnou strategii a implementuje v práci představenou metodu ASTUTE. Tato metoda vykazuje dobré detekční charakteristiky a je velmi jednoduše nastavitelná pomocí pouze jediného významného prahu. Její největší výhodou je, že poskytuje svoji detekční schopnost ihned po svém nasazení a tak je díky své povaze schopna okamžitě detekovat již probíhající anomálie, které by předchozí metody nebyly schopny odhalit kvůli své potřebě učit ze z historických. Její nevýhodou je ovšem horší detekční schopnost na sítích s velmi malým provozem. Protože zde v časových oknech není dostatečná mohutnost toků, vyvolává provoz několika málo webových klientů časté falešné poplachy na jednom z agregačních stupňů. Toto je možné samozřejmě zmírnit buď změnou agregace, zvýšením prahu pro detekci, případně dalším zpracováním.

Z celkového hlediska je však možné i přes výše popsané drobné problémy považovat metodu ASTUTE za úspěšnou a měla by být dále testována a případně nasazena do praktického provozu, kde by vzhledem ke svým vlastnostem a detekčním schopnostem byla jistě přínosem pro detekci anomálií v počítačových sítích. Implementace pluginů pro pohyblivé průměry ukázala, že i takto jednoduché metody mohou poskytovat dobré výsledky, ale velmi záleží na jejich pečlivém odladění pro konkrétní počítačovou síť.

Závěrem je třeba říci, že tyto metody sice vyžadují na počátku zvýšený vklad ve formě podrobnějšího studia problematiky a zapojení matematických postupů, ale získané výsledky jsou velmi slibné. Monitorování datové sítě pomocí statistických metod nad síťovými toky se tak zdá být velmi perspektivní možností jejich ochrany.

Literatura

[1] Arlt, J.; Arltová, M.; Rublíková, E.: *Analýza ekonomických časových řad s příklady*. Vysoká škola ekonomická, 2002 [cit. 2.4.2012], ISBN 80-245-0307-7, [online].
URL http://nb.vse.cz/~arltova/vyuka/crsbir02.pdf

[2] Beneš, V.; Prokešová, M.: Časoprostorové bodové procesy. In *ROBUST 2004*, Červen 2004 [cit. 2.4.2012], ISBN 80-7015-972-3, [online].
URL http://www.statspol.cz/robust/2004_robust2004.pdf

[3] Brownlee, N.; Mills, C.; Ruth, G.: Traffic Flow Measurement: Architecture. Říjen 1999 [cit. 11.11.2011], [online].
URL http://www.ietf.org/rfc/rfc2722.txt

[4] Claise, B.: Cisco Systems NetFlow Services Export Version 9. Říjen 2004 [cit. 11.11.2011], [online].
URL http://www.ietf.org/rfc/rfc3954.txt

[5] Daley, D.; Vere-Jones, D.: *An Introduction to the Theory of Point Processes, Volume I: Elementary Theory and Methods, Second Edition*. Springer, 2005, ISBN 978-0387955414.

[6] Danyliw, R.; Dougherty, C.; Shaffer, J.: Exploitation of vulnerability in SSH1 CRC-32 compensation attack detector. Listopad 2001 [cit. 5.1.2012], [online].
URL http://www.cert.org/incident_notes/IN-2001-12.html

[7] Dyken, C. E.: A Peek on Numerical Programming in Perl and Python. 2004 [cit. 28.10.2011], centre of Mathematics for Applications, University of Oslo, [online].
URL http://heim.ifi.uio.no/~erikd/pdf/hlarray.pdf

[8] EC-Council: *Ethical Hacking and Countermeasures: Threats and Defense Mechanisms*. Course Technology, 2009, ISBN 978-1435483613.

[9] Finch, T.: Incremental calculation of weighted mean and variance. Únor 2009 [cit. 3.1.2012], university of Cambridge Computing Service, [online].
URL
http://www-uxsup.csx.cam.ac.uk/~fanf2/hermes/doc/antiforgery/stats.pdf

[10] Golovanov, S.; Soumenkov, I.: TDL4 – Top Bot. 2011 [cit. 2.4.2012], [online].
URL http://www.securelist.com/en/analysis/204792180/TDL4_Top_Bot

[11] Handley, M.; Rescorla, E.: Internet Denial-of-Service Considerations. Listopad 2006 [cit. 3.1.2012], [online].
URL http://www.ietf.org/rfc/rfc3954.txt

[12] Hantzis, F.; Fyodor: Ncrack Reference Guide. [cit. 14.5.2012], [online].
URL http://nmap.org/ncrack/man.html

[13] Hipp, R. D.: Richard D. Hipp - osobní stránka. [cit. 2.4.2012], [online].
URL http://www.hwaci.com/drh/

[14] Kolektiv autorů: *The PDL Book*. 2012 [cit. 2.4.2012], draft [online].
URL http://pdl.perl.org/content/pdl-book-toc.html

[15] Lee, C. B.; Roedel, C.; Silenok, E.: Detection and Characterization of Port Scan Attacks. Technická zpráva, Department of Computer Science & Engineering University of California, San Diego, Kalifornie, USA, 2003 [cit. 5.1.2012], [online].
URL http://cseweb.ucsd.edu/~clbailey/PortScans.pdf

[16] Leonello, C.: jqPlot homepage. [cit. 17.5.2012], [online].
URL http://www.jqplot.com/

[17] Lyon, G.: Port Scanning Techniques. 2007 [cit. 5.1.2012], [online].
URL http://nmap.org/book/man-port-scanning-techniques.html

[18] McDowell, M.: Cyber Security Tip ST04-015, Understanding Denial-of-Service Attacks. 2004 [cit. 5.1.2012], [online].
URL http://www.us-cert.gov/cas/tips/ST04-015.html

[19] Namestnikov, Y.: The economics of Botnets. 2009 [cit. 2.4.2012], [online].
URL http://www.securelist.com/en/analysis/204792068/The_economics_of_Botnets

[20] Oetiker, T.: About RRDtool. Leden 2011 [cit. 3.1.2012], [online].
URL http://oss.oetiker.ch/rrdtool/

[21] Pras, A.; Sperotto, A.; Moura, G. C. M.; aj.: Attacks by "Anonymous" WikiLeaks Proponents not Anonymous. Technická zpráva, Design and Analysis of Communication Systems Group (DACS) University of Twente, Enschede, Holandsko, 2010 [cit. 25.4.2012], [online].
URL http://cseweb.ucsd.edu/~clbailey/PortScans.pdf

[22] Quittek, J.; Zseby, T.; Claise, B.; aj.: Requirements for IP Flow Information Export (IPFIX). Říjen 2004 [cit. 11.11.2011], [online].
URL http://www.ietf.org/rfc/rfc3917.txt

[23] RSnake; Lee, R. E.: Slowloris HTTP DoS. 2009 [cit. 30.3.2012], [online].
URL http://ha.ckers.org/slowloris/

[24] Silveiray, F.; Diot, C.; Taft, N.; aj.: ASTUTE: Detecting a Different Class of Traffic Anomalies. In *SIGCOMM Proceedings*, Září 2010 [cit. 17.12.2011], [online].
URL http://ccr.sigcomm.org/online/?q=node/651

[25] Turek, R.: Botnety. 2008 [cit. 2.4.2012], [online].
URL http://blog.synopsi.com/2008-04-27/botnety

[26] Čegan, J.: Ochrana datové sítě s využitím NetFlow dat. 2009 [cit. 30.3.2012], [online].
URL https://wis.fit.vutbr.cz/FIT/st/rp.php/rp/2008/BP/8087.pdf

[27] Čeleda, P.; Krejčí, R.; Vykopal, J.; aj.: Embedded Malware - An Analysis of the Chuck Norris Botnet. In *European Conference on Computer Network Defense*, 2010 [cit. 5.1.2012], ISBN 978-1-4244-9377-7, [online].
URL http://is.muni.cz/th/98863/fi_r/botnet-chuck-norris.pdf

[28] Vykopal, J.; Plesnik, T.; Minarik, P.: Network-Based Dictionary Attack Detection. In *Future Networks*, Březen 2009 [cit. 11.11.2011], ISBN 978-0-7695-3567-8, s. 23–27, [online].
URL http://is.muni.cz/th/51832/fi_r/dictionaryAttackDetection.pdf

[29] WWW stránky: Perl Data Language. 1999 [cit. 2.4.2012], [online].
URL http://sourceforge.net/projects/pdl/

[30] WWW stránky: NetFlow Performance Analysis. Květen 2007 [cit. 2.4.2012], [online].
URL http://www.cisco.com/en/US/technologies/tk543/tk812/technologies_white_paper0900aecd802a0eb9.html

[31] WWW stránky: Vulnerability Summary for CVE-1999-0153. 2008 [cit. 2.4.2012], [online].
URL http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-1999-0153

[32] WWW stránky: Nfsen-plugins. 2010 [cit. 4.2.2012], [online].
URL http://sourceforge.net/apps/trac/nfsen-plugins/

[33] WWW stránky: NetFlow Version 9 Flow-Record Format. Květen 2011 [cit. 20.12.2011], [online].
URL http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html

[34] WWW stránky: Netflow. Leden 2011 [cit. 3.1.2012], [online].
URL http://cs.wikipedia.org/wiki/Netflow

[35] WWW stránky: NFDUMP. Prosinec 2011 [cit. 3.1.2012], [online].
URL http://nfdump.sourceforge.net/

[36] WWW stránky: NfSen - Netflow Sensor. Prosinec 2011 [cit. 3.1.2012], [online].
URL http://nfsen.sourceforge.net/

[37] WWW stránky: Leet. Prosinec 2011 [cit. 5.1.2012], [online].
URL http://en.wikipedia.org/wiki/Leet

[38] WWW stránky: Cloud DDoS Protection. [cit. 2.4.2012], [online].
URL
http://www.imperva.com/products/wsc_cloud-ddos-protection-service.html

[39] WWW stránky: SQLite - stránky projektu. [cit. 2.4.2012], [online].
URL http://www.sqlite.org/

Příloha A

Obsah CD

- implementace pluginu pro klouzavý průměr
  - vlastní implementace pluginu
  - konfigurační údaje pro nfsen.conf
  - patch nfsenutil.php
- implementace pluginu pro vážený průměr
  - vlastní implementace pluginu
  - konfigurační údaje pro nfsen.conf
  - patch nfsenutil.php
- implementace pluginu pro metodu ASTUTE
  - vlastní implementace pluginu
  - konfigurační údaje pro nfsen.conf
  - patch nfsenutil.php
- skripty použité k analýze dat získané v průběhu monitorování
- zdrojové soubory této práce