VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DETEKCE APLIKACÍ PRO EFEKTIVNÍ SPRÁVU SÍTĚ POMOCÍ APPFLOW

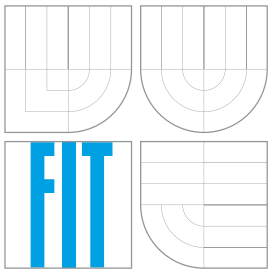
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUBOŠ NAVRÁTIL

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DETEKCE APLIKACÍ PRO EFEKTIVNÍ SPRÁVU SÍTĚ POMOCÍ APPFLOW

DETECTING NETWORK MANAGEMENT APPLICATIONS USING APPFLOW

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUBOŠ NAVRÁTIL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2013

Abstrakt

Tématem této bakalářské práce je detekování síťových aplikací pomocí technologie AppFlow, které napomáhá lepší informovanosti o monitorované síti. Cílem této práce je vytvoření rozšiřujícího pluginu pro zařízení FlowMon společnosti INVEA-TECH, který dokáže detekovat síťové aplikace a získané informace o těchto aplikacích exportovat pomocí protokolu IPFIX.

Abstract

This bachelor thesis is focused on the network application detection, using AppFlow in order to obtain information about a monitored network. The main goal of the thesis is to create plugin for FlowMon device that is made by INVIA-TECH. The plugin should provide a functionality for the detection of a network application and it should be also capable of exporting the detected information using the IPFIX protocol.

Klíčová slova

AppFlow, NetFlow, IP datový tok, protokol IPFIX, hloubkové analyzování paketů, vzory pro síťové aplikace

Keywords

AppFlow, NetFlow, IP data flow, protocol IPFIX, deep packet inspection, pattern for network applications

Citace

Luboš Navrátil: Detekce aplikací pro efektivní správu sítě pomocí AppFlow, bakalářská práce, Brno, FIT VUT v Brně, 2013

Detekce aplikací pro efektivní správu sítě pomocí AppFlow

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Matouška, Ph.D.

.....
Luboš Navrátil
13. května 2013

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé práce, Ing. Petru Matouškovi, Ph.D. a odborným konzultantům ze společnosti INVEA-TECH, Mgr. Martinu Elichovi a Ing. Petru Špringlovi, za jejich ochotu a věnovaný čas při řešení mé práce.

Obsah

Kapitola 1

Úvod

Datové sítě jsou v dnešní době díky rychlosti a značným možnostem využití nedílnou součástí civilizované společnosti. Jsou prostředkem pro získávání informací, běžné komunikace, ať už prostřednictvím elektronické pošty, hlasové komunikace s využitím VoIP (Voice over Internet Protocol - technologie pro přenos digitalizovaného hlasu prostřednictvím datové sítě) nebo využitím sociálních sítí, sdílení dat, dálkové ovládání jiných zařízení, řízení bankovních účtů až po zábavu a spoustu dalších rozmanitých využití.

Ruku v ruce s využíváním datových sítí jde *správa a monitorování datovýchsítí*. Toto odvětví provozování datové sítě se zabývá zvýšením bezpečnosti sítě, sledováním vytížení sítě a datového provozu v síti, to vše v reálném čase. V rámci bezpečnosti sítě se detekují různé anomálie v síťovém provozu a útoky, jako jsou například DOS/DDOS, SYN, SCAN a jiné. Z dlouhodobých statistik vytvořených sledováním sítě lze zjistit přetížení některých linek. Na základě tohoto zjištění může provozovatel sítě upravit její architekturu a tím zefektivnit síťovou komunikaci. Existují i požadavky na sledování aktivit uživatelů v síti. Některé společnosti monitorují aktivitu svých zaměstnanců na síti a mohou tak sledovat, jestli netráví pracovní dobu hraním her, sledováním multimedií nebo komunikací na sociálních sítích. Monitorování síťového provozu v reálném čase a ukládání dlouhodobé historie provozu umožňuje efektivní a rychlé řešení vznikajících problémů.

Tato práce se bude zabývat detekcí aplikací, které využívají monitorovanou síť, a exportováním informací o těchto aplikacích. Cílem práce je vytvoření rozšiřujícího pluginu pro sondy FlowMon společnosti INVEA-TECH, která bude rozeznávat síťové aplikace a výsledky monitorování odesílat na kolektor pomocí protokolu IPFIX. Rozeznávání paketů nebude založeno na kontrole zdrojových a cílových portů a použitého protokolu transportní vrstvy ISO/OSI, ale na porovnávání datové části paketu se vzory odpovídajícím jednotlivým aplikačním protokolům. Tento způsob hloubkového rozeznávání paketů se nazývá technologie AppFlow.

Text této práce je členěn do šesti kapitol. Po úvodní kapitole následuje kapitola zabývající se představením nejznámějších aplikací, které k dosažení svých cílů využívají technologii AppFlow. Třetí kapitola je zaměřena na bližší seznámení s technologií NetFlow, jeho rozšířením AppFlow a představením zařízení FlowMon společnosti INVEA-TECH. Ve čtvrté kapitole je poté popsán návrh řešení a implementace této práce do podoby rozšiřujícího pluginu pro sondy FlowMon. Následně navazuje pátá kapitola, ve které jsou uvedeny provedené testy vytvořeného pluginu a jejich výsledky. V závěrečné šesté kapitole se nachází zhodnocení celé práce a navržení možného pokračování v této práci.

Vytvořený rozšiřující plugin by měl výrazně pomoci při správě sítě a měl by vytvořit na první pohled konkrétnější představu o způsobu využití sítě uživateli.

Kapitola 2

Software pro detekci aplikací v síťovém provozu

V současné době již existují aplikace, které využívají AppFlow k dosažení svých cílů. V této kapitole jsou představeny některé z těch nejúspěšnějších a nejpoužívanějších aplikací, které demonstrují sílu a využitelnost technologie AppFlow v několika odlišných problematikách.

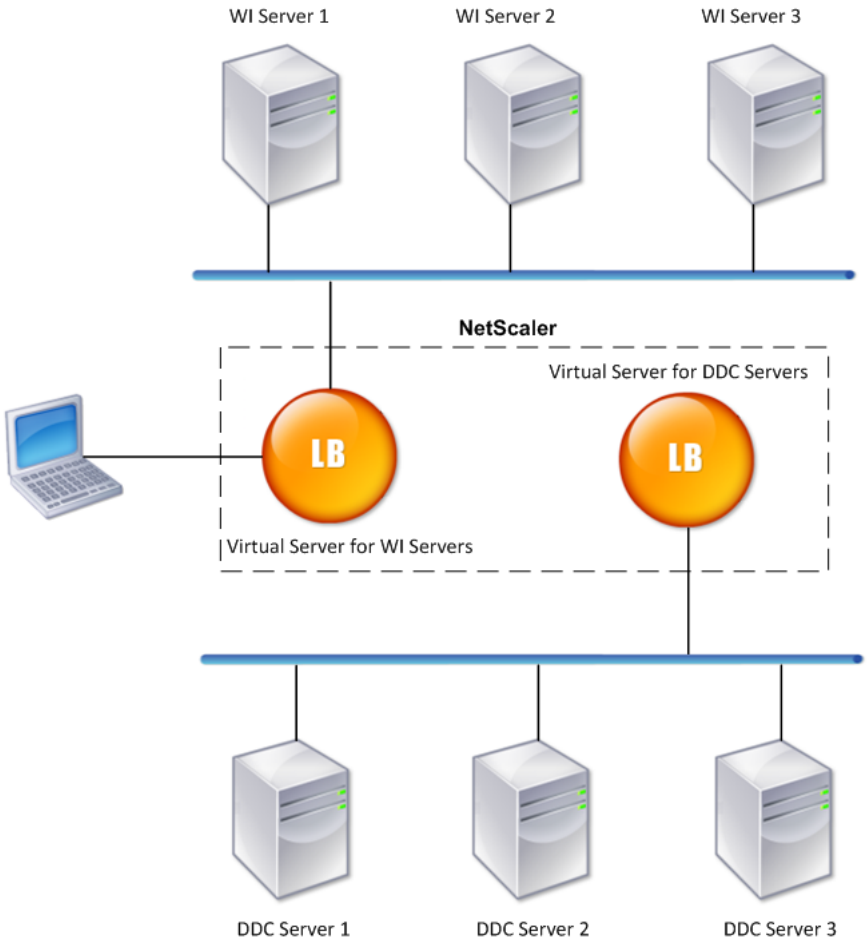
2.1 Citrix Netscaler 10

Citrix Netscaler 10[9] je systém, který se zaměřuje na zavádění *cloudu* do infrastruktury operátorů i podnikových sítí všech velikostí. Termínem *cloud* se označují služby, které nejsou v budovách ani pod správou organizace, ale jsou poskytovány organizaci třetí stranou, která služby udržuje a poskytuje prostřednictvím sítě za měsíční poplatek (těmito službami mohou být například office balíky, webová uložiště, mobilní bankovnictví atd.). *Citrix Netscaler 10* přináší integrované řešení, které poskytuje sítím všech velikostí pružnost, jednoduchost a rozšiřitelnost cloudu. To pomáhá dodávat zákazníkům na jakákoliv zařízení veřejné a privátní cloudové služby s optimálním výkonem, bezpečností a spolehlivostí.

AppFlow se v *Citrix Netscaler 10* využívá pro monitorování používaných aplikací klienty a stavu sítě. Shromažďuje podrobné informace o administrátorem zvolených aplikacích i jejich přenosových tocích a vytváří multidimenzionální pohledy na to, co se děje v reálném čase. Grafické a tabulkové zobrazení umožňuje okamžitě nalézt kritická místa v síti při poskytování aplikací klientům. Díky tomu může systém *load balancing*[10], který je součástí *Citrix Netscaler 10*, změnit v reálném čase politiku sítě a zachovat tak SLA (Service Level Agreement - dohoda o úrovni poskytování služeb) pro jednotlivé aplikace, viz. obrázek 2.1.

Při detekování nejnáročnějších aplikací je *Citrix Netscaler 10* schopný pomocí svého systému *ActionAnalytics* přesměrovat tyto uživatele na výkonnější server a tím zachovat SLA.

Systém je vhodný pro rozsáhlé společnosti i střední a malé podniky. Velké využití představuje také například pro poskytovatele internetu nebo jiných služeb. Kvalita softwaru a kvalitní podpora se ovšem odráží v ceně, která se pohybuje od 2000 dolarů do 30 000 dolarů v závislosti na rychlosti sítě a poskytovaných služeb v jednotlivých verzích.



Obrázek 2.1: Load Balancing (na obrázku LB) zajišťující SLA pro požadované aplikace, *Citrix Netscaler 10*. Převzato z [12]

2.2 StealthWatch

StealthWatch[11] je rozšíření pro systém *Citrix Netscaler* od firmy *Lancope*, monitoruje síť a hostitelské chování jako celku za účelem vytvoření celistvého pohledu na síť a rychlého varování na širokou škálu anomálii. Prostřednictvím sofistikované analýzy chování systém detekuje tzv. *zero-day* útoky (útoky využívající programátorské chyby v aplikaci před jejím odhalením a vytvořením opravy), vnitřní (insider) útoky (způsobené např. připojením zavirovaného PC do sítě, neoprávněný přístup, únik dat atd.) a mnoho dalších, se kterými se firemní sítě mohou setkat. Pomocí technologie AppFlow je schopný StealthWatch rychle identifikovat, které aplikace, technologie a uživatelé jsou příčinou útoků nebo poklesu výkonu sítě.

Systém je škálovatelný, aby vyhovoval potřebám i těch největších sítí a garantuje spolehlivou analýzu až do 3 miliónů toků za sekundu.

2.3 Free Real-Time AppFlow Analyzer

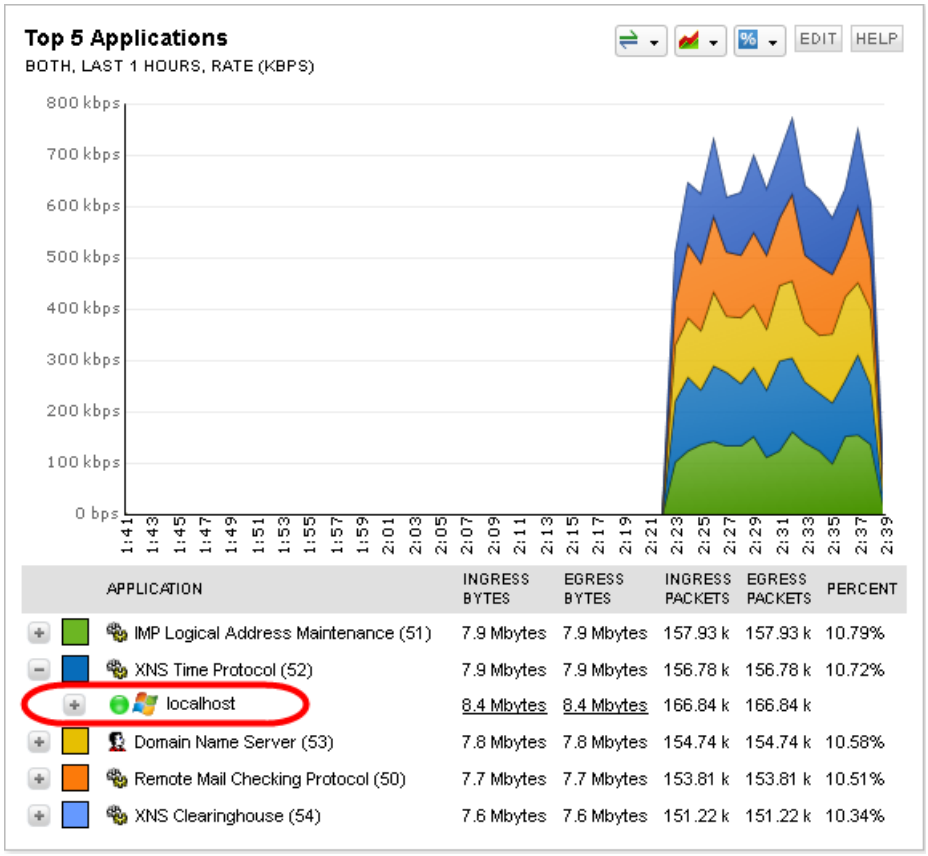
Tento bezplatný nástroj od firmy *Solarwinds* dokáže pouze přijímat a analyzovat záznamy nástrojů pro monitorování sítě na straně kolektoru. Je zde zmíněn zejména pro jeho propracovanou technikou pro vizualizaci AppFlow dat. Data obsahující záznamy AppFlow se mohou lišit v závislosti na použitých záznamových protokolech (většinou se vztahují ke společnostem, které vytvořily danou aplikaci). V tomto směru je *Free Real-Time AppFlow Analyzer* velmi flexibilní.

Free Real-Time AppFlow Analyzer[13] dokáže analyzovat záznamy typů:

- *Citrix NetScaler AppFlow* - výstupní záznam AppFlow analyzátoru firmu Citrix (Je součásti výše uvedeného *Citrix Netscaler 10*)
- *Juniper JFlow* - výstupní záznam technologie vzorkování IP provozu na směrovačích a přepínačích firmy *Juniper Networks*
- *sFlow* - výstupní záznam sFlow agentů, které můžeme nalézt jako součást ovladačů síťových rozhraní v přepínačích a směrovačích mnoha výrobců.
- *Cisco NetFlow* - standardní záznam NetFlow

Velmi podrobnou analýzou záznamů dokáže *Free Real-Time AppFlow Analyzer* rychle najít místa v síti, která jsou blokována nebo způsobují zpomalení síťového provozu. Dokáže také izolovat příchozí a odchozí provoz v komunikaci prostřednictvím sítě, umí zjistit protokoly, aplikace a domény, které byly v komunikaci použity, a umí identifikovat koncové body komunikace. Velmi povedenou vizualizací výsledků analýzy lze vypozorovat například, jaké aplikace jsou nejnáročnější na šířku přenosového pásma a kteří uživatelé tyto aplikace využívají, nebo v časových úsecích vypozorovat, v jaké době je síť nejvíce zatížená a co bylo příčinou zatížení.

Na obrázku 2.2 například vidíme prvních pět aplikací, které využívají největší šířku pásma. Kromě grafického znázornění se administrátorovi sítě zobrazí název aplikace, počet přijatých a odeslaných dat, počet přijatých a odeslaných paketů, ve kterých se data přenesla, a procentuální vyjádření využití sítě danou aplikací.



Obrázek 2.2: *Free Real-Time AppFlow Analyzer*. Převzato z [14]

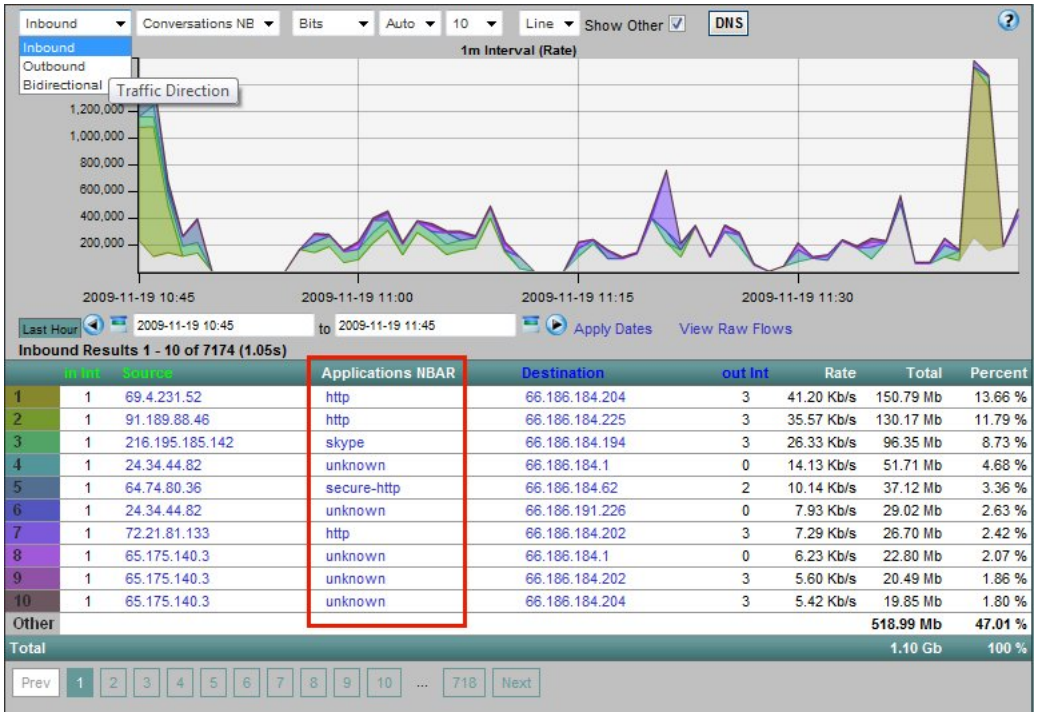
2.4 NBAR (Network Based Application Recognition)

Systém *NBAR*[15] od společnosti *Cisco* využívá rozeznávání aplikací pro klasifikaci aplikací do skupin a tím zajišťuje QoS(Quality of Service) - rezervace části dostupné přenosové kapacity sítě, aby při vytížení sítě nedocházelo ke snížení kvality síťových služeb. Jeho výhodou je, že se dá zapínat a vypínat za běhu *NetFlow Analyzeru* (kterého je součástí) z webového uživatelského rozhraní.

NBAR provádí hloubkovou inspekci paketů pro identifikaci používaných aplikací, proto je schopný rozeznat i aplikace jako *Skype*, *Kazaa* a další, které využívají pro komunikaci dynamické porty.

Na základě výsledků dělí *NBAR* aplikace do třech skupin. První skupinou jsou aplikace citlivé na zpoždění přenosu jako hlasové nebo multimediální služby přenášené v reálném čase. Do druhé skupiny jsou zařazeny aplikace, které vyžadují velkou šířku pásma a do třetí skupiny jsou zahrnuty aplikace, které jsou tolerantní ke ztrátě paketů při přenosu. Podle priorit je skupinám přidělováno přenosové pásmo a tím se zajišťuje QoS.

Na obrázku 2.3 vidíme uživatelské rozhraní systému *NBAR* kde je graficky znázorněno využití šířky přenosového pásma rozeznanými aplikacemi. Dále systém poskytuje podrobné informace o rozeznaných aplikacích, jako zdrojovou a cílovou IP adresu, název rozeznané aplikace, počet přenesených dat jednotlivými aplikacemi a procentuální vyjádření využití sítě danou aplikací.



Obrázek 2.3: *NBAR*, zobrazení rozeznaných aplikací, podrobných informací o aplikacích a využití šířky pásma v čase. Převzato z [15]

2.5 Shrnutí kapitoly

V této kapitole bylo prezentováno, v jakých problematikách je technologie AppFlow úspěšně využívána. Vzhledem k rozšiřování firemních (i jiných) sítí a přiklánění se k využívání tzv. cloudových aplikací se dá předpokládat, že se v nejbližší době stane nepostradatelnou součástí monitorovacích technik. Implementace technologie AppFlow je však velmi obtížná, a proto si ji jednotlivé firmy chrání a stává se jedním z jejich nejdražších "know-how".

V následující kapitole bude popsána technologie NetFlow a její rozšíření AppFlow.

Kapitola 3

Technologie NetFlow, rozšíření AppFlow a zařízení FlowMon

V této kapitole bude popsána funkčnost technologie NetFlow pro monitorování síťového provozu a jeho rozšíření AppFlow využívané pro detekci aplikací v síťovém provozu. Dále se seznámíme se zařízením *FlowMon*, pro které je tato práce implementována v podobě rozšiřujícího modulu.

3.1 NetFlow

Protokol NetFlow[17] byl původně vyvinutý společností Cisco jako rozšiřující služba Cisco směrovačů. Jeho úkolem je monitorování síťového provozu v reálném čase na základě *IP datových toků*. Protože monitorování síťového provozu je v dnešní době téměř nepostradatelnou součástí provozování a zabezpečení sítě, NetFlow si rychle získalo oblibu u:

- *Administrátorů sítí*
  - umožňuje celkový a podrobný přehled o provozu na jejich síti
  - umožňuje rychle detekovat vytížená místa v síti
  - umožňuje detekovat útoky na síť (jako např. DoS/DDoS)
- *Poskytovatelů internetových služeb*
  - účtování poplatků za služby
  - dodržování vyhlášky o elektronické komunikaci (ukládá provozovatelům veřejných komunikačních sítí povinnost uchovávat údaje o elektronické komunikaci)

Architektura NetFlow se skládá z:

- Exportéru
- Kolektoru

které jsou popsány v následujících podkapitolách.

3.1.1 NetFlow exportér

NetFlow exportér je připojen k monitorované lince sítě a analyzuje procházející pakety, které si spojuje do IP datových toků a na základě nich vytváří záznam obsahující statistiky, které obsahují například:

- začátek datového toku
- konec datového toku
- zdrojovou IP adresu
- cílovou IP adresu
- zdrojový port
- cílový port
- použitý protokol v komunikaci
- počet přenesených paketů

Vytvořené záznamy poté exportuje na kolektor. Pro export se používá NetFlow protokol. V současné době se nejvíce využívá protokol *v9*[2]. Struktura protokolu *v9* je založena na šablonách, což umožňuje jejich kombinacemi vytvářet různé záznamy *NetFlow*. Na základě protokolu *NetFlow v9* vznikl protokol *IPFIX* (Internet Protocol Flow Information eXport)[1], který je podporován směrovači mnoha předních výrobců síťových technologií a stal se standardem.

Pro přenos mezi exportérem a kolektorem používá NetFlow standardně UDP protokol. Po odeslání záznamu je z důvodu větší efektivity exportérem zahozen. Pokud se paket nepodaří doručit, není možnost ho znovu odeslat a je tedy trvale ztracen.

Netflow exportéry můžeme rozdělit na[3]:

- *Aktivní* - v pozici NetFlow exportérů jsou Cisco směrovače, které mimo směrování provozu provádí také výpočet NetFlow statistik
- *Pasivní* - zařízení specializovaná na monitorování síťového provozu a export NetFlow statistik

Výhody a nevýhody použití aktivních a pasivních exportérů jsou popsány níže v podkapitole o NetFlow architekturách.

3.1.2 NetFlow kolektor

NetFlow kolektor přijímá záznamy z NetFlow exportérů a ukládá je do dlouhodobých databází. Kolektor disponuje velkou kapacitou úložného prostoru. Součástí kolektoru bývají většinou aplikace, které zpracovávají data z kolektorových databází a zobrazují statistiky ve formě tabulek nebo grafů. To výrazně napomáhá k rychlejší analýze a nalezení problémů síťového provozu.

V architektuře NetFlow je možné exportovat data z několika exportérů (množina exportérů E) na několik kolektorů (množina kolektorů K), libovolně v poměru E:K. V praxi se však většinou používá architektura více exportérů (E) a jedním kolektorem, tedy v poměru E:1, kvůli jednodušším vyhodnocování dat (kolektory si mezi sebou nemusí přeposílat statistiky pro vyhodnocení celého síťového provozu).

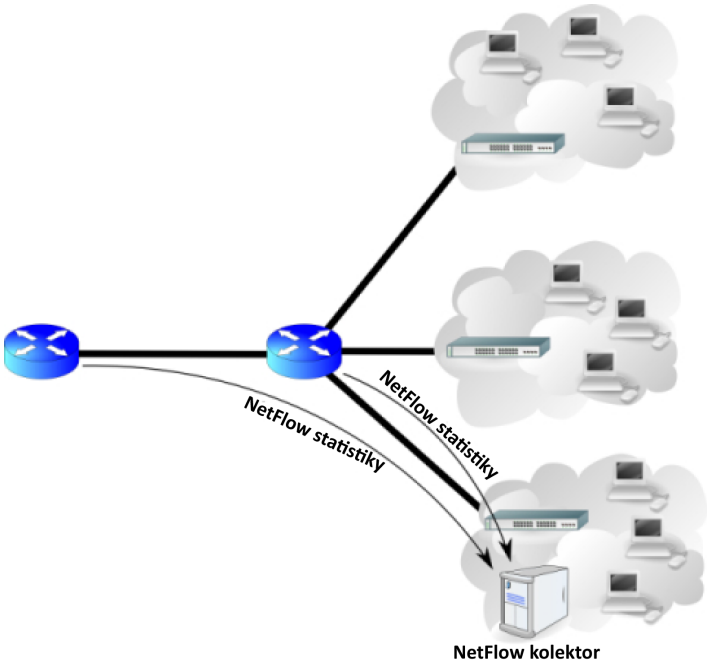
3.1.3 Architektura zapojení monitorované sítě pomocí NetFlow

Zde jsou popsány architektury zapojení sítí, v kterých jsou používány NetFlow exportéry a kolektory potřebné pro monitorování sítě. Jsou zde vysvětleny rozdíly mezi použitím aktivních a pasivních exportérů a důsledky, které má jejich použití na síťový provoz.

Původní architektura

Původní architektura navržená podle společnosti *Cisco* předpokládá použití *aktivních exportérů*, tedy směrovačů *Cisco*, které kromě směrování vypočítávají i NetFlow statistiky a exportují je na kolektor. Schéma zapojení podle původní architektury můžeme vidět na obrázku 3.1.

To má ovšem své nevýhody. Použitím *Cisco* směrovačů, které pracují zároveň jako NetFlow exportéry, se zvýší náklady na vytvoření sítě kvůli vysokým pořizovacím nákladům těchto směrovačů. Další významný problém představuje omezení směrovacího výkonu zařízení při výpočtu NetFlow statistik. Proto většina směrovačů s podporou NetFlow (s výjimkou těch nejdražších) využívá pro sběr statistik vzorkování. To znamená, že pro výpočet statistik se používá jen každý n-tý paket. To může způsobit i snížení pravděpodobnosti detekování útoků na síť.



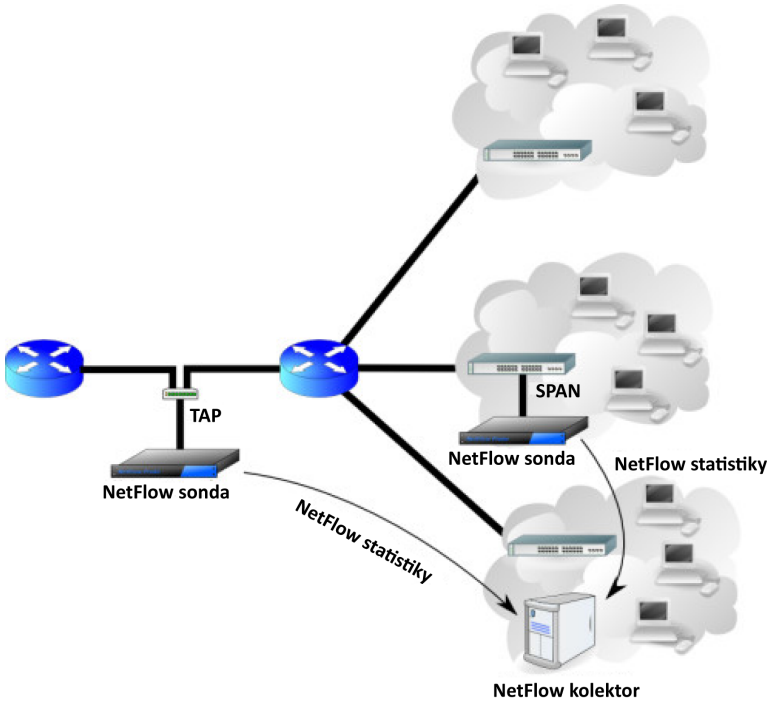
Obrázek 3.1: Původní architektura podle *Cisco*. Převzato z [16]

Moderní architektura

V současné době je oblíbenější a mnohem používanější architektura s *pasivními exportéry*. Pasivní exportéry jsou zařízení specializovaná na monitorování sítě, která jsou pro svoji jednoduchost levná a zároveň eliminují nevýhody *původní architektury*. Především při monitorování sítě procházející data pouze prohlížejí a nijak do nich nezasahují (proto pasivní

exportéry), tím pádem nijak neovlivňují rychlost ani plynulost průchodu dat na monitorované lince. Další výhodou oproti *původní architektuře* je, že *pasivní exportéry* lze umístit kamkoliv v síti, a tak i monitorovat jakékoliv místo v síti. Schéma zapojení podle moderní architektury můžeme vidět na obrázku 3.2.

Exportované statistiky jsou na kolektor odesílány dedikovanou linkou a to je dělá na monitorované lince neviditelnými (tím se účinně brání případným útočníkům).



Obrázek 3.2: Moderní architektura. Převzato z [16]

3.1.4 IP datový tok

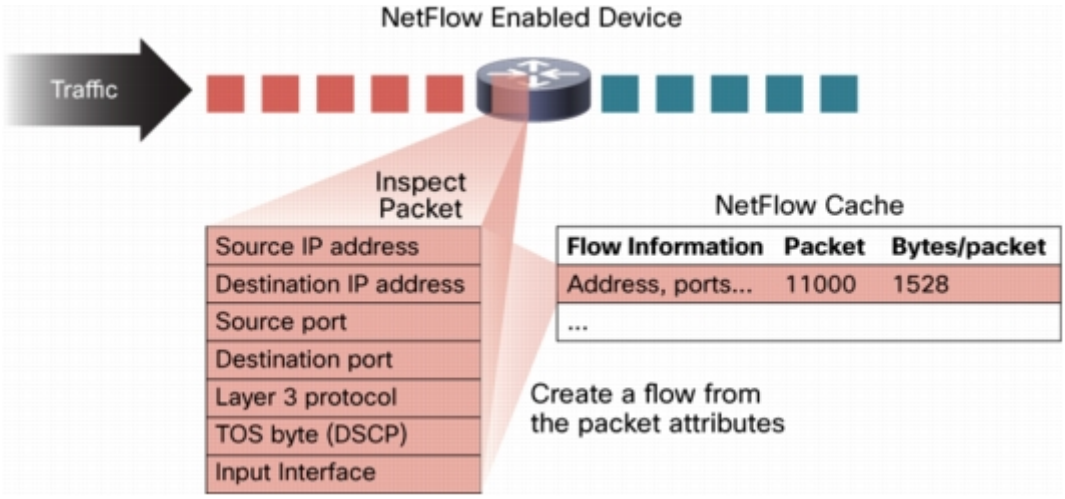
V průběhu této kapitoly byl již několikrát zmíněný termín *IP datový tok*[17]. Zde bude tato základní věc v monitorování sítě pomocí NetFlow vysvětlena.

IP datový tok je definovaný jako síťová konverzace se shodnými a v rámci toku neměnnými hodnotami(viz. obrázek 3.3):

- Zdrojové IP adresy
- Cílové IP adresy
- Zdrojového portu
- Cílového portu
- Přenosového protokolu transportní vrstvy ISO/OSI
- TOS (Type of Service) - určuje prioritu paketu při posílání sítí
- Vstupního rozhraní

Například navázání TCP spojení z pracovní stanice na webový server při čtení webové stránky je jeden tok. Začátek TCP spojení představuje začátek toku a uzavřením spojení dochází k ukončení IP toku. V průběhu TCP spojení se samozřejmě zdrojová/cílová IP adresa, zdrojový/cílový port ani vstupní rozhraní nemění.

Pro každý IP tok se zaznamenává čas jeho vzniku, doba trvání, počet přenesených paketů, objem přenesených dat a další požadované údaje.



Obrázek 3.3: IP datový tok. Převzato z [17]

3.1.5 Detekce aplikací pomocí NetFlow

Pomocí NetFlow jde teoreticky rozeznávat aplikace analýzou IP toků. Rozeznávání je založeno na sledování portů, na kterých komunikace probíhá, a použitého přenosového protokolu. Například přenos z webového serveru zpravidla probíhá na portu 80 a používá se přenosový protokol TCP.

Tento způsob je ale primitivní a má velmi malou pravděpodobnost úspěšného detekování aplikace. Důvodem je již dnes celkem běžná snaha o tunelování síťového provozu nežádoucími aplikacemi. Také existují aplikace, které využívají dynamického přidělování portu (např. komunikační program *Skype*), tudíž detekce těchto aplikací pomocí NetFlow je velmi obtížná a málo úspěšná.

V praxi se tento způsob detekce nepoužívá.

3.2 AppFlow

Technologie *AppFlow*[18] je rozšíření výše popsané technologie *NetFlow*. *AppFlow* je vznikající standard pro monitorování aplikací používaných v síťovém provozu a získávání podrobnějších informací o těchto aplikacích. Tvorba normy je otevřená široké veřejnosti, která se může k tvorbě normy vyjadřovat prostřednictvím webu *www.appflow.org*.

AppFlow je technologie založená na hloubkovém analyzování obsahu paketu. Tudíž pro detekci aplikací nepotřebuje znát čísla využívaných portů, ani transportní protokol. Analýza paketu spočívá v oddělení části paketu, která nese datový obsah, od hlavičky paketu.

Ta část paketu, která obsahuje data, je podrobena porovnávání se vzory uloženými v databázi (většinou ve formě regulárních výrazů). Pokud nastane shoda, celý tok, kterého je tento paket součástí, se prohlásí za datový tok používaný danou aplikací (viz vlastnosti IP datového toku).

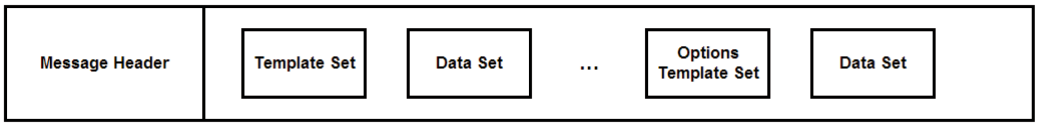
Z toho vyplývá, že základ pro úspěšné detekování aplikace závisí na správnosti a aktuálnosti vzorů používaných pro porovnávání.

3.2.1 Protokol IPFIX

AppFlow využívá pro export dat protokol *IPFIX*[1]. *IPFIX* je standardem skupiny IETF (Internet Engineering Task Force - Komise techniky Internetu), vycházející z protokolu *NetFlow v9*, zajišťující exportování dat z exportéru na kolektor pomocí jednotného přenosového mechanismu, formátu dat a zabezpečení. *IPFIX* je založen na šablonách, tzv. *FlowSet*. FlowSety mohou být třech typů:

- *Data Set* - šablona obsahující přenášená data
- *Template Set* - definice nově vytvořené šablony (musí se odesílat jako první, hned jak je to možné, jinak kolektor neporozumí datům v *Data Set*)
- *Option Template Set* - doplňující nastavení přikládané k *Data Set* a *Template Set*

FlowSety se mohou v *IPFIX* paketu v libovolných kombinacích a množství (omezení je pouze maximální velikost paketu).



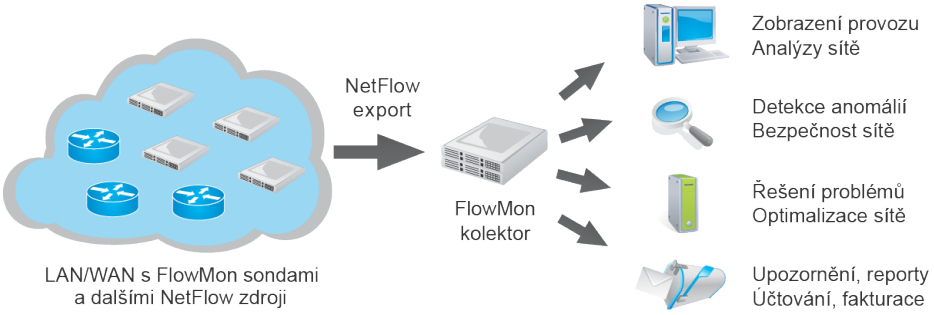
Obrázek 3.4: IPFIX paket

3.3 Zařízení FlowMon

Zařízení *FlowMon*[19] společnosti *INVEA-TECH* umožňuje monitorování síťového provozu v reálném čase, dohled nad síťovými prvky a službami, zvýšení bezpečnosti počítačové sítě odhalením vnějších i vnitřních útoků, rychlé a efektivní řešení problémů v síti, účtování a fakturaci na základě přenesených dat a mnoho dalších možností užitečných pro pohodlnou a plnohodnotnou správu sítě.

Řešení *FlowMon* zahrnuje, jak můžeme vidět na obrázku 3.5, výkonné autonomní sondy exportující statistiky o síťovém provozu na monitorované síti, kolektory pro analýzu těchto statistik a jejich vizualizaci a moduly, které rozšiřují použitelnost *FlowMon* (jako například modul pro dohled nad sítí a jejími službami, detekce anomálií, pokročilejší vizualizaci síťových statistik atd.).

Díky využití průmyslového standardu *NetFlow* je systém jednoduše rozšiřitelný, kompatibilní s produkty třetích stran a velmi dobře škálovatelný.



Obrázek 3.5: FlowMon architektura. Převzato z [19]

3.3.1 FlowMon sondy

FlowMon sondy jsou pasivní monitorovací zařízení určená pro ethernetové sítě na rychlostech od 10 Mb/s do 10 Gb/s. Sondy monitorují komunikaci na síti, vytvářejí statistiky, které jsou plně kompatibilní s *NetFlow* standardem a odesílají je na vestavěný či externí kolektor. Pro běžné sítě *FlowMon* nabízí standardní modely sond, pro vysoce vytížené linky nabízí hardwarově akcelerované modely sond.

Pro menší a střední sítě a pro rychlé seznámení s technologií obsahují sondy vestavěný kolektor umožňující sběr, analýzu a vizualizaci síťových statistik. Pro použití ve větších sítích jsou pro sběr dat z více sond použity samostatné kolektory.

3.3.2 FlowMon kolektory

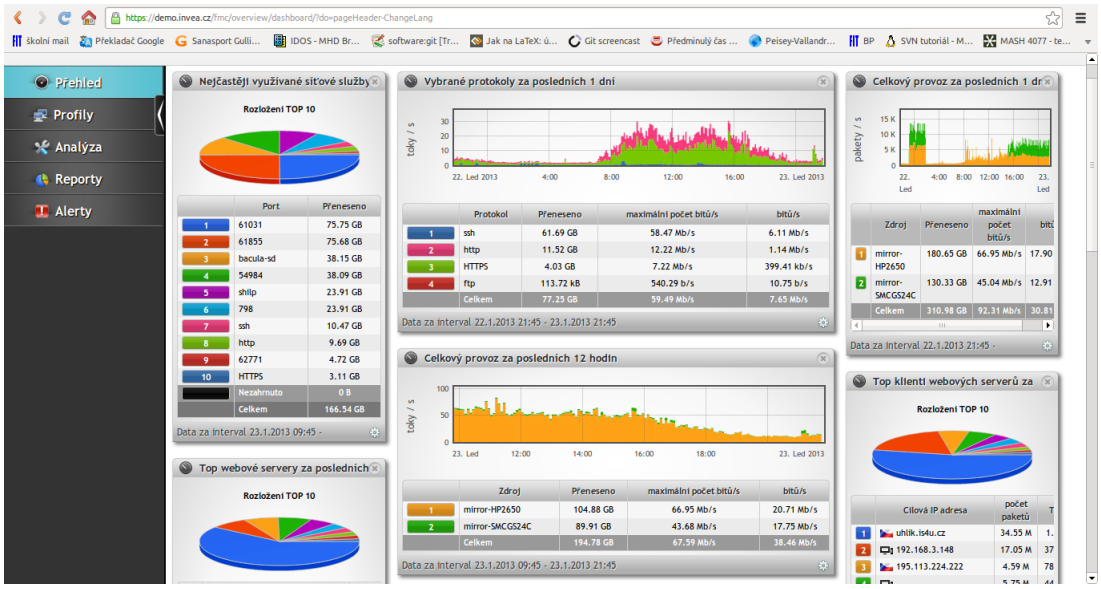
Kolektory jsou zařízení s vysokou síťovou kapacitou primárně určená pro sběr, uložení a vizualizaci síťových statistik exportovaných sondami. Zobrazení uložených *NetFlow* dat a jejich analýzy probíhají prostřednictvím zabezpečeného webového rozhraní.

3.3.3 Rozšiřující moduly

Funkcionalitu systému lze rozšířit pomocí rozšiřujících modulů (takzvaných *pluginů*) určených pro sondy *FlowMon*. Pluginy se dělí do čtyř základních skupin podle jejich použití a to na vstupní (*input plugin*), procesní (*process plugin*), filtrovací (*filtr plugin*) a exportní (*export plugin*). Na obrázku 3.6 můžeme vidět webové rozhraní, kde se zobrazují statistiky získané sondou *FlowMon*. Mezi nejznámější a nejvyužívanější pluginy patří:

- nástroje pro dohled nad servery a síťovými službami
- automatická detekce anomálií v síti
- pokročilá vizualizace NetFlow statistik pomocí grafů komunikací
- detekce neoprávněných přístupových bodů k internetu
- měření odezev kritických serverů a služeb

I tato práce je vyvíjena jako rozšiřující modul (plugin) pro sondy technologie FlowMon.



Obrázek 3.6: FlowMon webové rozhraní pro vizualizaci statistik

3.4 Shrnutí kapitoly

V této kapitole jsme se seznámily s technologií NetFlow, kterou využívá zařízení *FlowMon* společnosti INVEA-TECH pro monitorování sítě. Dále byla popsána technologie AppFlow, která je rozšířením technologie NetFlow a zabývá se detekcí aplikací. Technologie AppFlow je využita při implementaci detekce aplikací v podobě rozšiřujícího modulu sondy *FlowMon*. Návrh a implementace celé práce je popsán v následující kapitole.

Kapitola 4

Návrh řešení a implementace

Cílem této práce je vytvoření pluginu pro technologii *FlowMon*, která detekuje základní aplikace a informace o nich používané v síťovém provozu, a následně exportuje tyto statistiky prostřednictvím protokolu *IPFIX* na kolektor. V návrhu řešení je nutné zaměřit se především na správné a efektivní detekování aplikací, což znamená důraz na správné oddělení datové části paketu a zaručení co nejmenšího počtu časově náročných porovnávání. Dále je nutné navrhnout šablonu pro *IPFIX*, která bude obsahovat užitečné informace pro administrátory sítě, kteří budou tento plugin využívat.

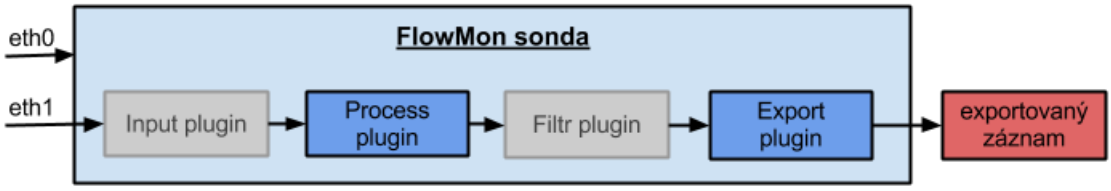
4.1 Virtuální FlowMon sonda

Pro vývoj pluginu pro *FlowMon* mi byla poskytnuta virtuální *FlowMon sonda verze 5.03* s exportérem *flowmonexp verze 3.02.11*. Ke zprovoznění FlowMon sondy jsem použil freeware virtualizační program pro simulaci hardwarového prostředí *VirtualBox v4.2.4*. Na Flowmon sondě jsou spuštěny dvě rozhraní:

- *eth0* - rozhraní je určeno pro administraci sondy
- *eth1* - monitorovací rozhraní

Pro pohodlnější administraci a práci s FlowMon sondou je možné použít protokol pro vzdálený přístup SSH. Další firewallem sondy povolené protokoly pro vzdálený přístup jsou HTTP, HTTPS, SNMP a Zabbix.

Obrázek 4.1 zobrazuje blokové schéma *FlowMon* sondy, kde můžeme vidět pořadí vyhodnocování dat monitorované sítě prostřednictvím pluginů. FlowMon sonda pracuje se



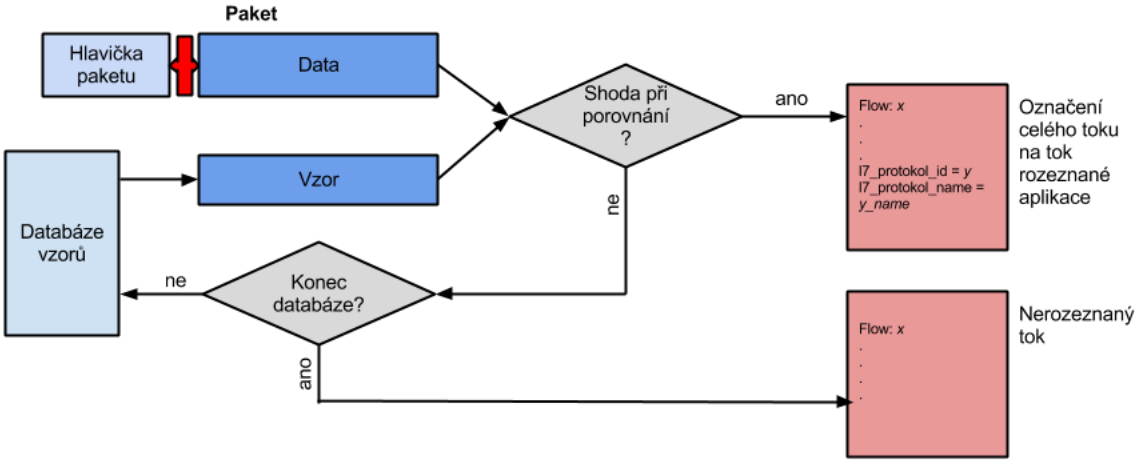
Obrázek 4.1: Blokové schéma FlowMon sondy

čtyřmi typy pluginů. Pro funkčnost sondy je nutné použít některý z *input pluginů* a *output pluginů*, které mohou být doplněny použitím *process pluginu* a *filtr pluginu*. Každý plugin se skládá z pro sebe specifických funkcí, které tvoří kostru každého pluginu a řídí jeho funkčnost. Implementace této práce je rozdělena do dvou pluginů. V *process pluginu* je implementovaná část rozeznávání aplikací, zatímco export prostřednictvím IPFIX protokolu je implementován v *export pluginu*.

4.2 Technologie rozpoznávání aplikací

Jak už bylo zmíněno dříve, rozpoznávání aplikací v této práci není založeno na analýze informací získaných z hlavičky paketu, jako jsou zdrojový a cílový port, a použitý přenosový protokol transportní vrstvy ISO/OSI. Rozpoznávání aplikací je provedeno technologií AppFlow, která je založena na hloubkovém analyzování paketu[5].

Prvním krokem při rozeznávání aplikací technologií AppFlow je oddělení datové části rozpoznávaného paketu od hlavičky paketu. Dále se z databáze vzorů, která je přiložena k vytvořenému pluginu a ve které jsou uloženy vzory jednotlivých síťových aplikací, vybere vzor protokolu síťové aplikace a porovná se s datovou částí paketu. Pokud dojde ke shodě, paket je označen za paket síťové aplikace daného vzoru a s ním je stejně označen celý tok, do kterého paket patří (označení celého toku bude vysvětleno níže v sekci 4.3 o process pluginu). Pokud nenastane shoda, vybere se vzor další síťové aplikace z databáze a porovnávání se opakuje dokud se nenarazí na konec databáze vzorů. Pokud při porovnávání nenastane shoda a zároveň se narazí na konec databáze vzorů, do záznamu o toku nebudou přidány položky s informacemi o síťových aplikacích (prostřednictvím validační funkce, blíže v sekci 4.5 o vytvoření záznamu). Celý způsob rozeznávání je naznačen na obrázku 4.2. Implementace porovnávacích funkcí a funkcí starajících se o načítání vzorů jsem obdržel od firmy INVEA-TECH jako základ ke své práci.



Obrázek 4.2: Blokové schéma technologie rozeznávání síťových aplikací

Vzory použité pro rozeznávání síťových aplikací mají formu regulárních výrazů. Kvalita regulárních výrazů a míra, do jaké dokáží rozeznávat pakety v průběhu komunikace síťových aplikací, mají největší vliv na úspěšnost rozeznání síťových aplikací. Zároveň je ale velice

obtížné je vytvořit. Pro zlepšení úspěšnosti rozeznání často používaného HTTP protokolu jsem, pomocí *RFC 2616*[4] a databáze filtrů aplikace *Snort*, vytvořil regulární výraz, který zahrnuje velkou část formátů paketu HTTP protokolu. Tento regulární výraz je společně s ostatními testován a výsledky jsou zobrazeny v kapitole testování. Původní HTTP vzor má tvar:

```
http/(0\.9|1\.0|1\.1) [1-5][0-9][0-9]|post [\x09-\x0d -~]* http/[01]
\.[019]
```

Nově vytvořený vzor má tvar:

```
(GET|POST|HEAD|OPTIONS|PUT|DELETE|TRACE) /.* ?(HTTP/[01]\.[019])?\s(
Host:|Content-Length:|Content-Type:|Connection:|Referer:)
```

Při vytváření vzoru jsem se zaměřil na tvar existujících HTTP dotazů a položek, které mohou obsahovat. Snahou bylo, aby regulární výraz pokryl co největší počet HTTP dotazů používaných v běžném síťovém provozu. Přesto že jsem vytvořil tento regulární výraz, tvorba vzorů síťových aplikací není cílem této práce.

Databáze regulárních výrazů je tvořena soubory s příponou `*.pat`, které obsahují vzory síťových aplikací. Soubory se vzory jsem převzal z webové stránky *protocolinfo.org*, kde komunita zabývající se technologií AppFlow tyto vzory vytváří a volně poskytuje. Každý vzor síťové aplikace je uložen v samostatném souboru. To je výhodné pro úpravy a aktualizace jednotlivých vzorů nezávisle na vytvořeném pluginu. U souboru se musí dodržet určitý formát pro správné načtení vzorů. Řádek v souboru `*.pat` začínající znakem `#` je považován za komentář a není vyhodnocován. Na prvním neprázdném řádku, který není komentář, musí být název protokolu (např. HTTP) a na následujícím neprázdném řádku musí být vzor ve formě POSIX regulárního výrazu. Příklad souboru `*.pat` se vzorem síťové aplikace:

```
#
# Zde mohou být komentáře
#

# název protokolu
ftp
# vzor ve formě POSIX regulárního výrazu
^220[\x09-\x0d -~\x80-\xfd]*ftp
```

Seznam vzorů aplikací, které mají být použité pro rozeznávání jsou uložené v konfiguračním souboru s příponou `*.config`, který se předává jako povinný parametr pluginu. V konfiguračním souboru je uloženo na jednotlivých řádcích id vzoru a úplná cesta k souboru se vzorem aplikačního protokolu, oddělených od sebe znakem ':'. Výhodou konfiguračního souboru je rychlý výběr vzorů, které chceme použít pro rozeznávání síťových aplikací nezávisle na vytvořeném pluginu. Příklad konfiguračního souboru `*.config`:

```
1:patterns/protocols/http.pat
3:patterns/protocols/irc.pat
4:patterns/protocols/ftp.pat
5:patterns/protocols/dns.pat
6:patterns/protocols/dhcp.pat
```

4.3 Process plugin

Část práce, která se zabývá detekcí aplikací, je implementována jako *process plugin* sondy FlowMon. Pro funkčnost process pluginu je nutné spustit jej vždy v kombinaci s *input pluginem* využívajícím funkci `unsigned char *plugin_input_get_packet()`, která je jednou ze čtyř povinně volitelných funkcí kostry *input pluginu*. Tato jediná funkce předává z *input pluginu* do *process pluginu* celý paket, což je nutné k provádění detekce síťových aplikací v *process pluginu*.

Process plugin je tvořen, kromě inicializační funkce `plugin_process_init()`, dalšími třemi základními funkcemi. Tyto funkce se provádí podle situace, která nastane po obdržení paketu do *process pluginu*. Funkce `plugin_process_create()` se provede v případě, že *process pluginu* byl předán paket, který byl vyhodnocen jako první paket nového toku. Funkce `plugin_process_update()` se provede v případě, že *process pluginu* byl předán paket, který je součástí již existujícího toku. Funkce `plugin_process_release()` se provede v případě, že zachycený tok je uvolněn z *process pluginu* a vytvořený záznam je předán *export pluginu*.

Přiřazení paketů do příslušného toku je prováděno na základě vypočítaných hašů (*hash funkce* - algoritmus pro převod vstupních dat do relativně malého čísla), které jsou vypočítány ihned při příjmu paketu do *input pluginu*. *Hash funkce* je součástí vnitřní implementace FlowMon sondy. Výpočet haše zahrnuje základní informace z hlavičky paketu:

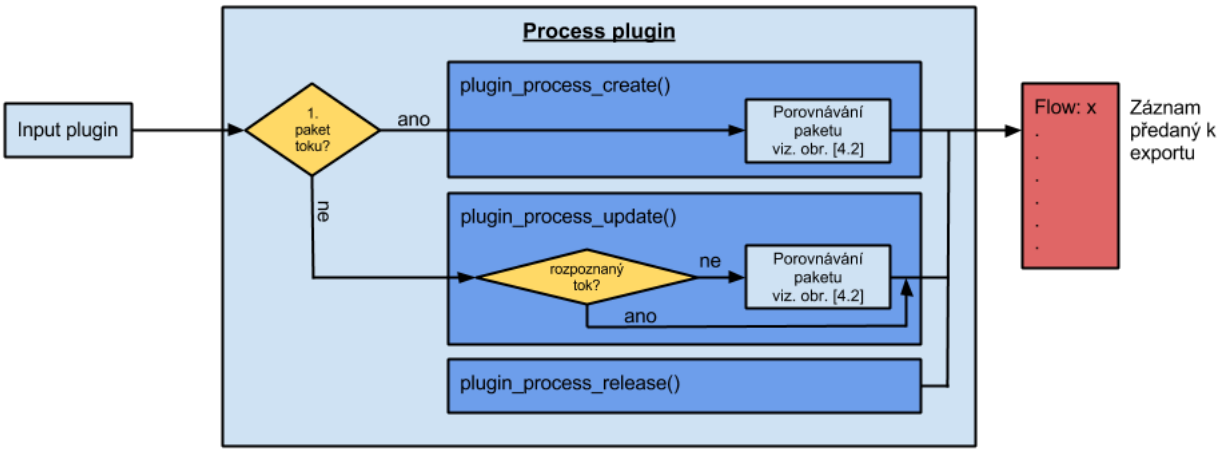
- zdrojovou a cílovou IP adresu
- zdrojový a cílový port
- přenosový protokol transportní vrstvy ISO/OSI
- vstupní rozhraní

Vytváření toků je ovšem také ovlivněno vnitřním nastavením parametrů sondy a to:

- maximálním počtem paketů v toku
- časovým úsekem, po kterém musí být tok exportován

Funkce vytvořeného *process pluginu* je zobrazena na obrázku 4.3. Při příchodu paketu, který je vyhodnocen jako první paket toku, dojde k zavolání funkce `plugin_process_create()` a v této funkci dojde k porovnání datové části paketu se vzory síťových aplikací (viz. obrázek 4.2) a následného doplnění informací o rozeznaném toku. Při příchodu paketu, který je vyhodnocen jako paket již existujícího toku, dojde k zavolání funkce `plugin_process_update()` a v této funkci dojde k porovnání datové části paketu se vzory síťových aplikací, pouze když tok ještě nebyl označen za tok nějaké síťové aplikace (žádný předchozí paket patřící do stejného toku ještě nebyl rozeznán), jinak se paket porovnávat nebude a pouze se zvýší počítadlo paketů patřících do daného toku. Při rozeznání toku síťové aplikace tedy mohou nastat dvě krajní situace. V nejlepším možném případě dojde k rozeznání při porovnání prvního paketu. V této situaci bude rozeznaný paket i všechny následující pakety, které budou patřit do tohoto toku, označeny za pakety dané síťové aplikace a počet porovnávání pro tento tok bude roven jedné. V nejhorším možném případě se tok rozezná až při porovnání posledního paketu patřícího do daného toku. V této situaci bude tento paket a i všechny předchozí pakety patřící do tohoto toku označeny za pakety dané aplikace a

počet porovnávání pro tento tok bude *N*M*, kde *N* je počet paketů v toku a *M* je počet vzorů. Využití schopnosti process pluginu zařadit příchozí pakety do příslušného toku tedy výrazně zvyšuje efektivitu detekce síťových aplikací.



Obrázek 4.3: Blokové schéma funkčnosti *process pluginu*

4.4 Doplňující informace o rozeznaných síťových aplikacích

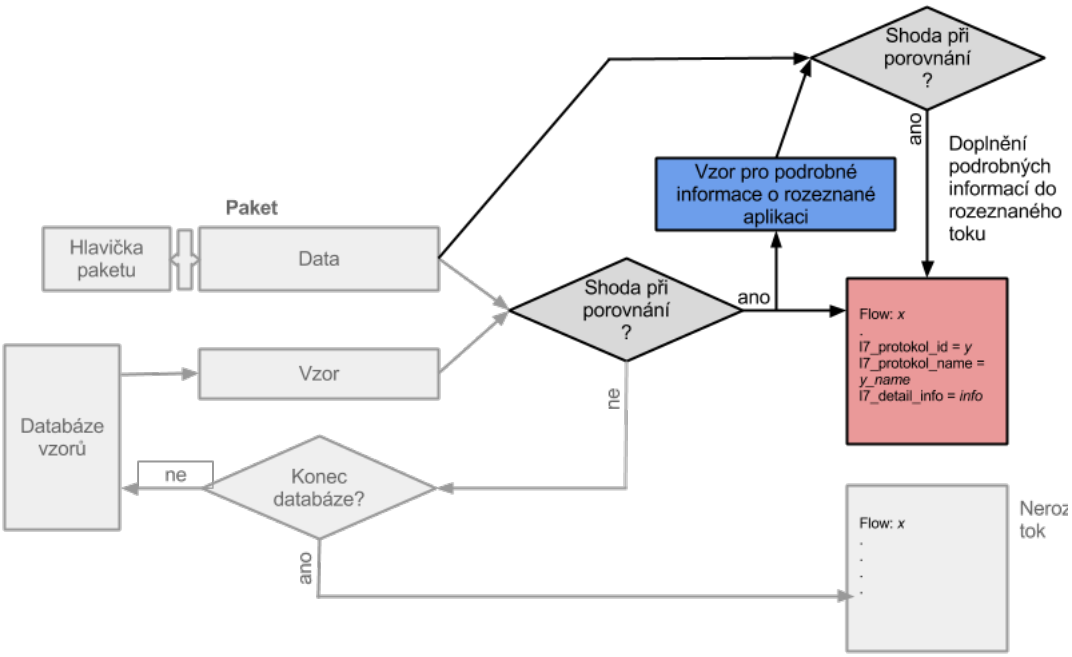
Za účelem získání více informací a tím i zlepšení představy o dění na monitorované síti jsem vytvořil funkce pro získávání doplňujících informací o rozeznaných síťových aplikacích. Stejně jako u rozeznávání aplikací je i získávání doplňujících informací založeno na porovnávání datové části paketu s určitým vzorem v podobě regulárního výrazu. Zde jsou ovšem použity vzory vytvořené pro získávání doplňujících informací. Na rozdíl od rozeznávání aplikací je zde využita struktura `regmatch_t`, pomocí které lze získat požadovanou část odpovídající regulárnímu výrazu. Pro získání požadované části z regulárního výrazu stačí obalit tuto část do kulatých závorek. Pro příklad je zde uveden regulární výraz pro zjištění hodnoty položky *host:* z paketu protokolu HTTP:

```
\nhost: ([^\n]*)\n
```

Nastane-li shoda při porovnávání vzoru s datovou částí paketu, do položek struktury `regmatch_t` se uloží offset začátku a konce uzávorkované části. Poté se pomocí těchto offsetů získá pouze požadovaná část datové části paketu.

Získávání podrobných informací se provádí pouze u rozpoznaných toků. Pokud už v daném toku byly doplňující informace získány, jsou doplněny do záznamu pro export a dále se pro tento tok porovnávání za účelem získání doplňujících informací neprovádí (využití funkcí *process pluginu* [4.3]).

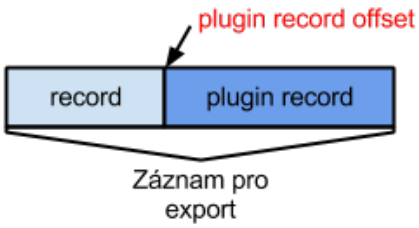
Funkce je vytvořena tak, aby tvorba vzorů pro získávání doplňujících informací byla co nejjednodušší. Ve většině případů lze využít vzory pro detekci aplikací a pouze uzávorkovat část vzoru, která bude odpovídat požadovaným informacím. Pro prezentaci funkčnosti jsem vytvořil vzory pro získávání doplňujících informací o protokolu HTTP a to *absolutní cestu k souboru na serveru* používanou při HTTP dotazu GET a položku *Host*, ve které je uvedeno doménové jméno.



Obrázek 4.4: Blokové schéma získávání podrobných informací o rozeznaných síťových aplikacích (navazuje na obrázek 4.2)

4.5 Vytvoření záznamu předávaného pro export

Informace o síťových tocích získané analýzou paketu, detekováním síťových aplikací a získáním doplňujících informací o detekovaných aplikacích se ukládají do uceleného záznamu, který je předán pro export *export pluginu*. Záznam je tvořen ze dvou částí. První část záznamu, část *record*, je typu `flow_record_t` a obsahuje základních NetFlow informace, které jsou získány z hlavičky paketu. Část *record* se vytvoří ihned po obdržení paketu do *input pluginu* a jeho vytvoření se provádí automaticky. Druhá část záznamu, část *plugin record*, je složena z položek záznamu vytvořených v *process pluginu*. Položky obsahují informace o detekovaných síťových aplikacích a doplňující informace vztahující se k těmto aplikacím. Položky v záznamu `plugin record` jsou vytvořeny pomocí takzvaných *getterů*. *Getter* je struktura, která obsahuje popis a hodnotu položky vytvářeného záznamu.



Obrázek 4.5: Záznam pro export

Vytvoření části záznamu *plugin record* je provedeno ve funkci `plugin_process_getter_init()`, kde je každá položka, kterou chceme předat k exportu, vytvořena funkcí `getter_add()`. Typ každé položky záznamu (od teď už jen *getter*) určuje sedm parametrů funkce `getter_add()`:

- `flow_record_getter_t ** getter_list` - ukazatel na seznam gettrů, do kterého bude vytvořený getter přidán.
- `char *name` - jméno vytvořeného gettru.
- `int length` - standardní délka gettru v bytech. Pokud je hodnota -1, znamená to, že délka gettru je proměnlivá.
- `void *self` - ukazatel na data daného gettru.
- `int (*valid)()` - validační funkce, v *process pluginu* definována jako `RECORD_VALID(name) int name()`. Tato funkce rozhoduje o tom, zda bude getter přidán do celkového záznamu a bude exportován. Pokud funkce vrátí hodnotu nula, getter nebude přidán do záznamu. Pokud funkce vrátí hodnotu jedna, getter bude přidán do celkového záznamu a bude exportován.
- `int (*current_length)()` - funkce, v *process pluginu* definována jako `RECORD_CURRENT_LENGTH(name) int name()`, která vrací pravou délku daného gettru v bytech. Tato funkce je velmi důležitá u gettrů s proměnnou délkou.
- `void (*filler)()` - plnící funkce, v *process pluginu* definována jako `RECORD_FILLER(name) void name()`. Tato funkce naplní vytvořený getter odpovídající hodnotou.

Validační funkci `RECORD_VALID()`, funkci pro zjištění pravé délky gettru `RECORD_CURRENT_LENGTH()` a plnící funkci `RECORD_FILLER()` by měl mít každý getter vlastní. Společné funkce se mohou použít pouze za předpokladu, že bude zachovaná validita gettru.

4.6 Export záznamů prostřednictvím protokolu IPFIX

Pro export záznamů prostřednictvím protokolu IPFIX jsem využil *FlowMon IPFIX Export Plugin*[8] vytvořený Mgr. Petrem Velanem. *FlowMon IPFIX Export Plugin* umožňuje export záznamů o tocích ve formátu IPFIX pomocí UDP, TCP nebo SCTP protokolu. Je založen na specifikaci protokolu IPFIX (RFC 5101)[1] a podporuje prvky definované společnostmi (pod vlastním *enterprise ID*) a také proměnné délky prvků. Plugin zajišťuje, že každý záznam o toku je přiřazen k odpovídající IPFIX šabloně, takže všechny dostupné prvky (s validační hodnotou 1) jsou exportovány na kolektor.

Pro příjem exportovaných dat z exportéru jsem použil kolektor *Ipfixcol*[1], který je určen pro příjem exportovaných dat prostřednictvím IPFIX protokolu a program *fbitdump*, který je určen pro výpis přijatých dat *Ipfixcol* kolektorem. Soubory, které jsou nutné upravit na straně kolektoru pro správný příjem a interpretaci dat, se tedy vztahují právě ke kolektoru *Ipfixcol* a programu *fbitdump*.

[1] https://www.liberouter.org/ipfixcol/

Strana exportéru

Aby byl zaručen správný export, musela být na straně exportéru vytvořena IPFIX šablona[7]. Šablona je vytvořena v souboru `/etc/flowmon/ipfix-template-file.txt` a musí obsahovat definice prvků záznamu, které jsou určeny pro export. Prvky záznamu jsou v souboru šablony uloženy v určitém formátu. Každý prvek je definován na jednom řádku, přičemž jednotlivé informace o prvku jsou od sebe odděleny tabulátorem. Jako první je uvedeno *jméno prvku* (gettru). Následuje *enterprise ID*, tedy identifikační číslo společnosti, která prvek vytvořila. Pokud *enterprise ID* je '0', znamená to, že prvek je jedním ze standardních prvků pro export pomocí protokolu IPFIX definovaných společností IANA (Internet Assigned Numbers Authority)[6]. Jako třetí je uvedeno *element ID*, tedy číslo prvku podle standardu společnosti IANA nebo společnosti, která prvek vytvořila (vztahuje se k *enterprise ID*). Jako poslední je uvedena délka prvku v bytech. Pokud je délka prvku '-1', prvek je definován s proměnnou délkou. Šablona použitá pro export má tvar:

```
# GETTER_NAME       Enterprise ID    Element ID    Length

FLOW_START_MSEC           0              154         8
FLOW_END_MSEC             0              155         8
PACKETS                   0              2           8
INPUT_INTERFACE           0              10          2
OUTPUT_INTERFACE          0              14          2
L3_PROTO                  0              60          1
L3_IPV4_ADDR_SRC          0              8           4
L3_IPV4_ADDR_DST          0              12          4
L3_IPV4_TOS               0              5           1
L3_IPV6_ADDR_SRC          0              27          16
L3_IPV6_ADDR_DST          0              28          16
L4_PROTO                  0              4           1
L4_TCP_FLAGS              0              6           1
L4_PORT_SRC               0              7           2
L4_PORT_DST               0              11          2
L7_PROTOCOL_ID            0              95          2
L7_PROTOCOL_NAME          0              96          -1
L7_HTTP_GET_PATH          4193           200         -1
L7_HTTP_HOST              4193           201         -1
```

Strana kolektoru

Na straně kolektoru je nezbytné upravit soubory, které obsahují informace potřebné k správné interpretaci dat. Soubor `/etc/ipfixcol/ipfix-elements.xml` je nutné rozšířit o popis prvků, které jsou definovány v šabloně na straně exportéru. Náhled souboru je zobrazen na obrázku 4.6. Soubor je ve formátu xml, kde všechny definované prvky jsou obaleny základním párovým elementem `ipfix-elements`. Definice jednotlivých prvků jsou obsaženy v párovém elementu `element`. Položkami definice prvku jsou elementy:

- `enterprise` - jeho hodnota odpovídá hodnotě `enterprise ID` v šabloně na straně exportéru.
- `id` - jeho hodnota odpovídá položce `element ID` v šabloně na straně exportéru

- `name` - zde je uvedeno jméno exportovaného prvku podle standardu společnosti IANA, případně podle společnosti, která prvek vytvořila.
- `dataType` - zde je uveden typ exportovaného prvku.
- `semantic` - zde je uvedeno jakého významu prvek je a jak bude interpretován.

ipfix-elements.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ipfix-elements>
        <element>
                <enterprise>0</enterprise>
                <id>152</id>
                <name>flowStartMilliseconds</name>
                <dataType>dateTimeMilliseconds</dataType>
                <semantic></semantic>
        </element>
        <element>
                <enterprise>0</enterprise>
                <id>153</id>
                <name>flowEndMilliseconds</name>
                <dataType>dateTimeMilliseconds</dataType>
                <semantic></semantic>
        </element>
        <element>
                <enterprise>0</enterprise>
                <id>10</id>
                <name>ingressInterface</name>
                <dataType>unsigned32</dataType>
                <semantic>identifier</semantic>
        </element>
        <element>
                <enterprise>0</enterprise>
                <id>14</id>
                <name>egressInterface</name>
                <dataType>unsigned32</dataType>
                <semantic>identifier</semantic>
        </element>
        <element>
                <enterprise>0</enterprise>
                <id>60</id>
                <name>ipVersion</name>
                <dataType>unsigned8</dataType>
                <semantic>identifier</semantic>
        </element>
```

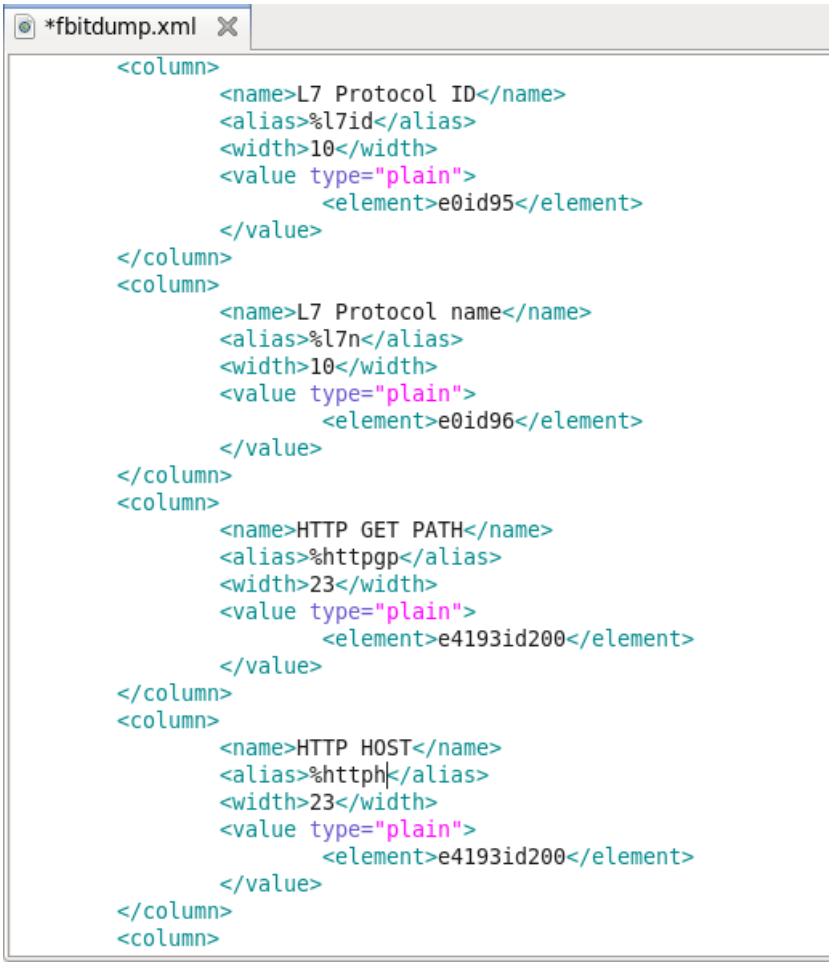
Obrázek 4.6: Soubor ipfix-elements.xml

Druhým souborem, který je nutné upravit na straně kolektoru, je soubor `/usr/share/fbitdump/fbitdump.xml`, který obsahuje definice pro zobrazování prvků. Náhled souboru je zobrazen na obrázku 4.7. Soubor je také ve formátu xml. Definice prvků jsou zahrnuty v základním párovém elementu s názvem `columns`. Definice jednotlivých prvků jsou poté uvedeny uvnitř párového elementu `column`. Položkami definice prvku jsou elementy:

- `name` - zde je uveden název prvku tak jak bude zobrazen při výpisu exportovaných dat na straně kolektoru.
- `alias` - zde je uvedena zkratka daného prvku začínající znakem `%`. Tato zkratka se používá při výběru a stanovení pořadí prvků, které mají být zobrazeny ze souboru

přijatých dat, pomocí programu *fbitdump* (zkratky se používají jako parametry programu *fbitdump*).

- `width` - udává místo v počtech znaků, které je rezervováno pro prvek daného typu (pomyslná šířka sloupce).
- `value` - určuje hodnotu prvku. U elementu `value` lze použít parametr *type*, pomocí kterého lze definovat typ hodnoty prvku. Hodnota je do elementu `value` přidávána pomocí dalšího elementu `element`.
  - `element` - hodnota elementu má vždy tvar `e`x`id`y, kde x udává enterprise ID daného prvku a y udává element ID daného prvku, určeného standardem společnosti IANA nebo společností, která daný prvek vytvořila. Párový element `element` má také parametry *semantics* a *aggregation* pomocí kterých lze blíže určit jakým způsobem a v jakém tvaru mají být data interpretována (např. příznak pro výpis IP adresy).



Obrázek 4.7: Soubor fbitdump.xml

4.7 Shrnutí kapitoly

V této kapitole byla podrobněji představena *FlowMon sonda* a její funkční části použité při implementaci této práce. Dále zde byla popsána technologie rozpoznávání aplikací a získávání podrobnějších informací o rozpoznaných aplikacích, které využívají pro větší efektivitu přednosti *process pluginu*. Byl zde představen formát vzorů pro rozpoznávání aplikací a také vytvořený vzor HTTP, který byl vytvořen za účelem zvýšení úspěšnosti rozpoznání protokolu HTTP v síťovém provozu. Také zde byl představen formát záznamu určených k exportu, způsob jeho vytvoření a potřebné kroky k úspěšnému exportu prostřednictvím protokolu IPFIX ze strany exportéru a správnému přijmutí a interpretaci exportovaných záznamů na straně kolektoru.

Následující kapitola se bude zabývat testováním vytvořeného pluginu, vzorů pro detekci síťových aplikací a exportu získaných informací na kolektor.

Kapitola 5

Testování

Testování vytvořeného pluginu bylo rozděleno na dvě části. V první části testování jsem se zaměřil na zjištění úspěšnosti detekování aplikací. Tato část testování probíhala na virtuální sondě, která mi byla poskytnuta, a byla využita oklasifikovaná data společnosti ISCX (information security centre of excellence) *testbed-12Jun.pcap*. Součástí testovacích dat v podobě pcap souboru s názvem *testbed-12Jun.pcap* je soubor *testbed-12Jun.xlsx*, kde je pcap soubor rozdělen do IP datových toků. O každém IP datovém toku jsou uvedeny podrobné informace, včetně názvu síťové aplikace a počtu paketů, které do těchto toků patří. Testování detekce aplikací zahrnovalo jaký vliv má mnou vytvořený vzoru HTTP na úspěšnost detekování tohoto protokolu a úspěšnost detekování vybraných síťových protokolů. V druhé části jsem se zaměřil na zatížení sondy při použití pluginu. Toto testování probíhalo na sondě FlowMon v laboratoři FIT VUT.

5.1 Testování nového vzoru protokolu HTTP

Testování vzoru HTTP probíhalo na virtuální FlowMon sondě. Jako testovací data jsem použil oklasifikovaná data v podobě pcap souboru s názvem *testbed-12Jun.pcap*.

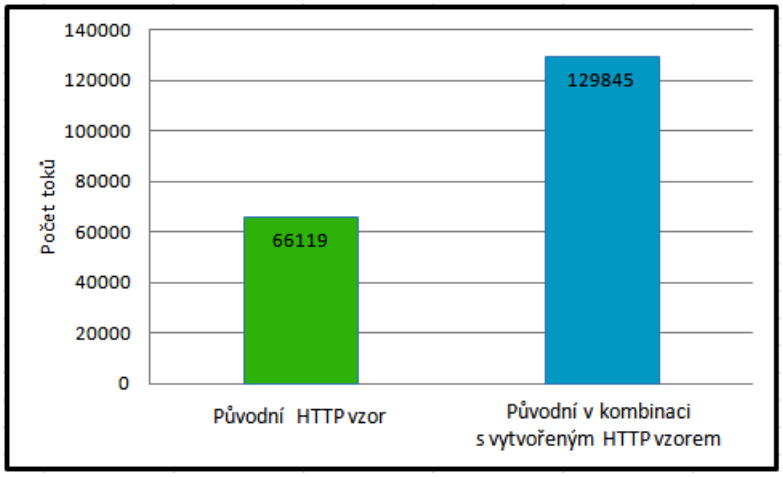
V testovacím souboru je celkem 5 121 051 paketů protokolu HTTP obsažených v 85 520 IP datových tocích. Nejprve nad těmito testovacími daty proběhla detekce protokolu HTTP pouze s původním vzorem převzatým z *protocolinfo.org*. Poté byl k původnímu vzoru přidán i nově vytvořený vzor protokolu HTTP a s kombinací těchto vzorů proběhl test znovu. Výsledky testu jsou zobrazeny v tabulce 5.1.

testbed-12jun.pcap			
	Detekce pomocí HTTP vzoru		Skutečný počet HTTP
	Původní	Původní v kombinaci s vytvořeným	
Počet toků	66119	129845	
Počet paketů	2956012	4899079	5121051

Tabulka 5.1: Tabulka zobrazující výsledky testu nového vzoru protokolu HTTP

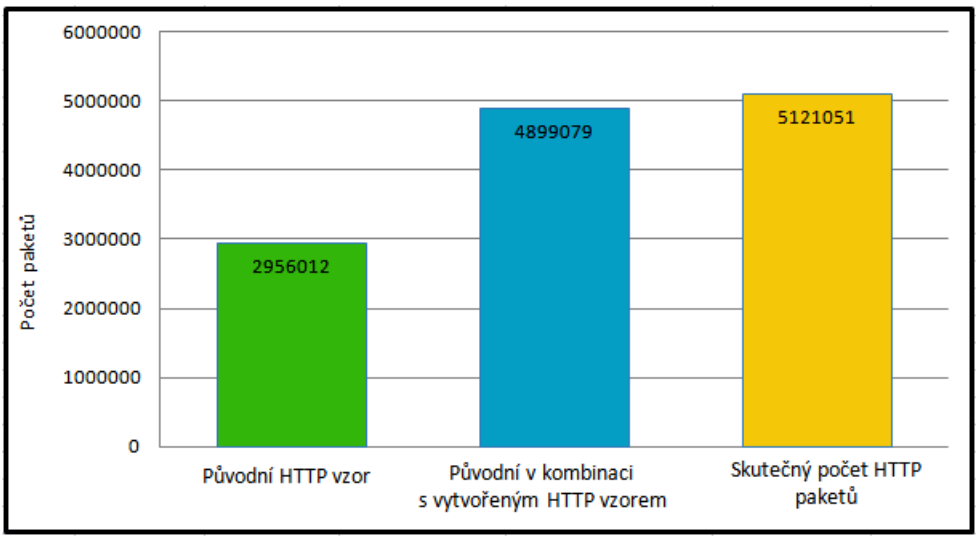
V grafu na obrázku 5.1 je viditelný nárůst rozeznaných IP datových toků protokolu HTTP. Z důvodu různých podmínek použitých pro vytváření IP datových toků uvedených

v referenčním souboru *testbed-12Jun.xlsx*[1] a podmínek použitých při vytváření IP datových toků na sondě FlowMon, které se mohou lišit například v maximálním počtu paketů v jednom toku nebo maximálním časovým intervalem pro export toku, nelze porovnávat počet rozeznaných IP datových toků na sondě FlowMon s počtem IP datových toků uvedených v referenčním souboru *testbed-12Jun.xlsx*. Proto je tento graf pouze orientační.



Obrázek 5.1: Graf znázorňující počet rozeznaných toků protokolu HTTP

V grafu na obrázku 5.2 je zobrazen počet rozeznaných paketů protokolu HTTP původním vzorem (zelený sloupec) a kombinací původního a nově vytvořeného vzoru (modrý sloupec) a skutečný počet paketů protokolu HTTP v testovacích datech.



Obrázek 5.2: Graf znázorňující počet rozeznaných paketů protokolu HTTP a skutečný počet paketů protokolu HTTP v testovacích datech

Při detekci protokolu HTTP původním vzorem bylo rozpoznáno *57,72%* skutečného

[1] http://www.iscx.ca/datasets

počtu paketů protokolu HTTP. Při detekci kombinací původního vzoru s nově vytvořeným vzorem bylo rozpoznáno *95,66%* skutečného počtu paketů HTTP protokolu. Přidání nově vytvořeného vzoru tedy zvýšilo úspěšnost rozeznání protokolu HTTP o *37,94%*.

Důležitým zjištěním je, že v průběhu testu nebyl za paket protokolu HTTP chybně označen žádný paket jiného protokolu.

5.2 Testování vzorů pro vybrané síťové protokoly

Testování vzorů pro vybrané síťové protokoly probíhalo stejně jako u testování nového vzoru pro protokol HTTP na virtuální sondě FlowMon s využitím testovacích dat *testbed-12Jun.pcap*, které obsahují celkem 5 960 892 paketů.

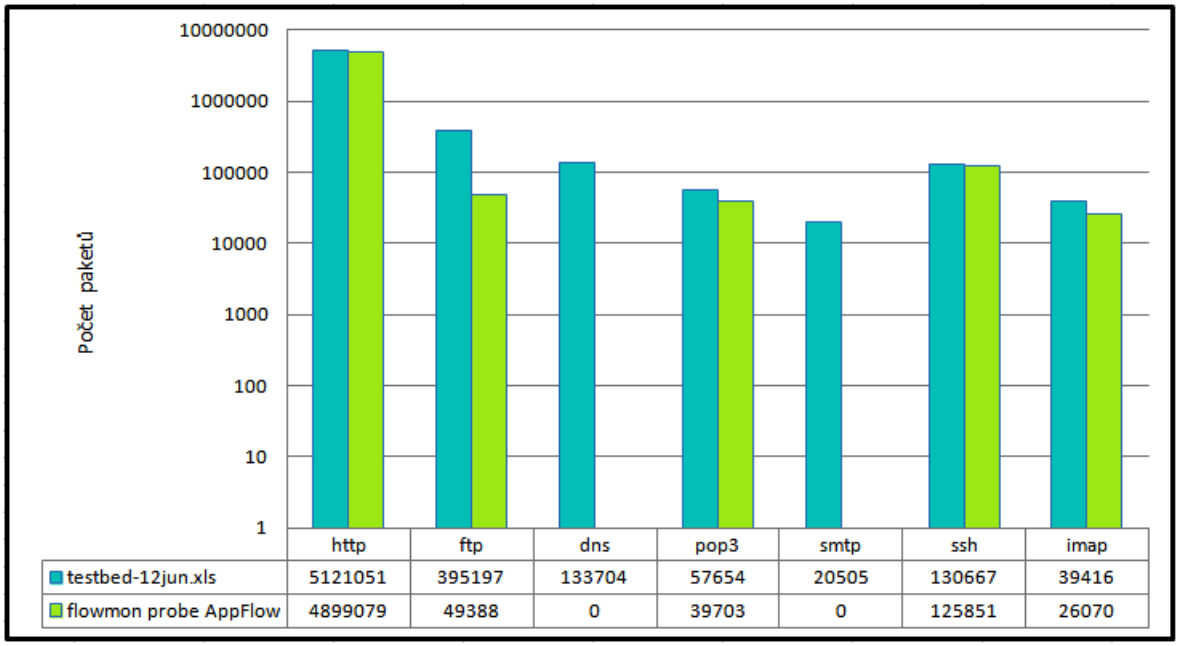
V tabulce 5.2 je uvedeno sedm základních a často používaných síťových protokolů. Pro každý protokol jsou uvedeny informace o počtu toků a paketů, které jsou obsaženy v testovacích datech a počtu toků a paketů, které byly rozpoznány na sondě FlowMon pomocí vytvořeného pluginu. Kvůli výše uvedeným rozdílům mezi referenčními daty a FlowMon sondou při vytváření IP datových toků, není možné počty referenčních a rozpoznaných IP datových toků porovnávat. Úspěšnost detekce síťových aplikací (protokolů) je tedy zjištěna na základě počtu rozpoznaných paketů.

testbed-12jun.pcap					
	Testovací sada		Rozpoznané pomocí pluginu		
Protokol	Toků	Paketů	Toků	Paketů	Úspěšnost rozpoznání v %
http	85520	5121051	132299	4899079	95,66%
ftp	1955	395197	230	49388	12,49%
dns	32238	133704	0	0	0%
pop3	2095	57654	2098	39703	68,86%
smtp	315	20505	0	0	0%
ssh	2649	130667	5063	125851	96,31%
imap	4156	39416	623	26070	66,14%

Tabulka 5.2: Tabulka referenčních dat a dat získaných testováním vytvořeného pluginu FlowMon sondy

Na obrázku 5.3 je zobrazen graf znázorňující úspěšnost detekce jednotlivých síťových protokolů. Pro každý protokol je v modrém sloupci zobrazen skutečný počet paketů daného protokolu, který je obsažen v testovacích datech a v zeleném sloupci počet paketů, které byly rozpoznány jako pakety daného protokolu vytvořeným pluginem sondy FlowMon. Pro rozeznávání protokolu zde byl použit původní vzor v kombinaci s nově vytvořeným vzorem.

Testováním bylo dokázáno, že detekce síťových aplikací prostřednictvím vytvořeného pluginu sondy FlowMon je funkční. Největší vliv na úspěšnost rozpoznání síťových aplikací mají vzory pro jednotlivé aplikace. Některé vzory, například pro HTTP nebo SSH protokoly, prokázaly vysokou úspěšnost rozeznávání daného protokolu a bylo by možné je využít i v praxi. Naopak vzory pro DNS, SMTP a FTP protokoly kvůli své nízké úspěšnosti rozpoznávání nejsou vhodné pro nasazení do praxe a je nutné je upravit.



Obrázek 5.3: Graf znázorňující počet rozeznaných paketů sedmi základních protokolů a skutečný počet paketů těchto protokolů v testovacích datech

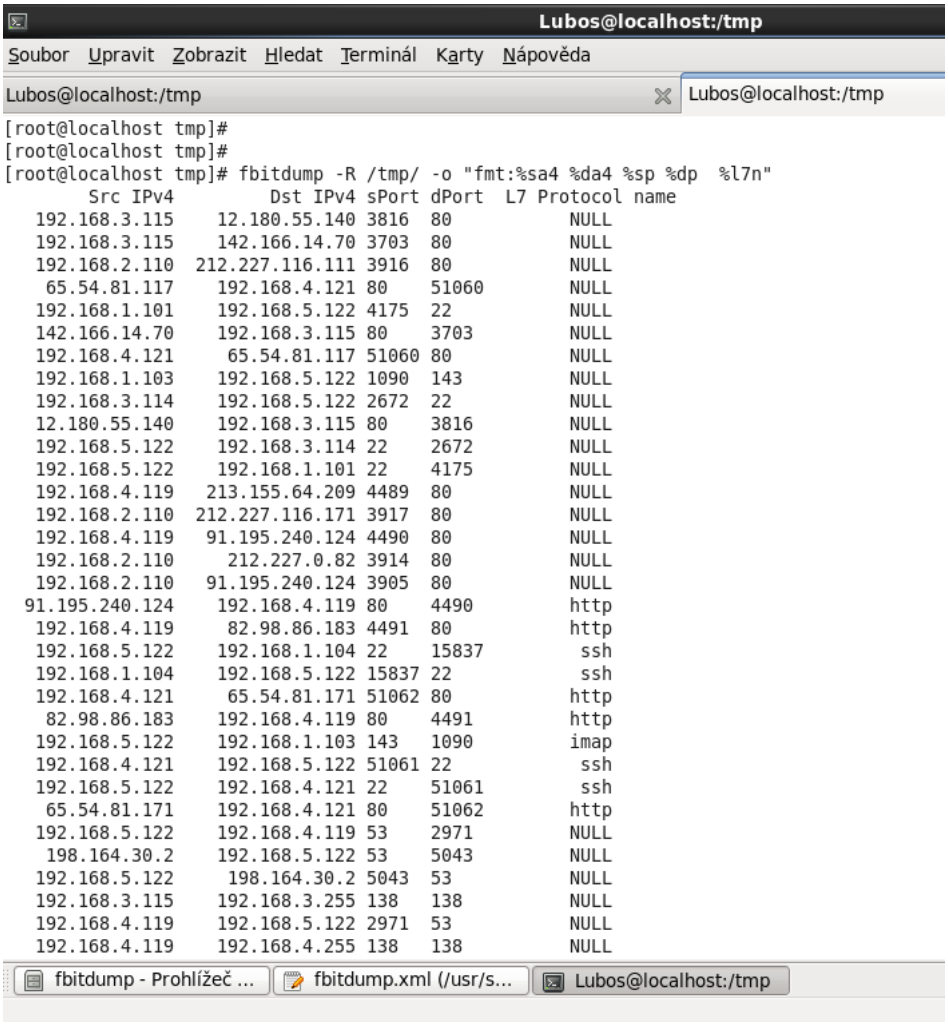
5.3 Testování zatížení sondy a exportu prostřednictvím protokolu IPFIX

Test zatížení sondy probíhal na FlowMon sondě umístěné v laboratoři FIT VUT a byl rozdělen do dvou částí. V první části jsem podrobil sondu zátěžovému testu, kdy jsem pomocí programu *tcpreplay* s parametrem *–topspeed* (pakety se odesílají tak rychle jak je to jen možné - *man tcpreplay*) posílal pakety vysokou frekvencí na monitorovací rozhraní sondy. V extrémním zátěžovém testu zatížení sondy nepřesáhlo *42%*.

V druhé části byla sonda připojena do sítě VUT s běžným provozem po dobu 1 hodiny. Při běžném provozu zatížení sondy nepřesáhlo *16%*.

Export prostřednictvím protokolu IPFIX

Exportovaná data prostřednictvím protokolu IPFIX byla protokolem TCP přenesena na kolektor *Ipfixcol* nainstalovaný na virtuálním stroji se systémem *Scientific Linux 6.4*. *Ipfixcol* ukládá přijatá data ve formátu *fastbit*. Pro čtení přijatých dat byl použit program *fbitdump*, pomocí kterého byla data zobrazována na standardní výstup bez grafického rozhraní ve formátu jednoduché tabulky. Na obrázku 5.4 je zobrazen výpis přijatých dat na kolektor programem *fbitdump*. Formát výstupu je zadán při spuštění v parametrech programu *fbitdump*. Zde je vidět výpis dat ve formátu zdrojová a cílová IP adresa, zdrojový a cílový port a jméno rozeznané síťové aplikace (pokud síťová aplikace nebyla rozeznána, je na výstupu hodnota *NULL*)



Obrázek 5.4: Výpis exportovaných dat programem fbitdump

5.4 Shrnutí kapitoly

Tato kapitola se zabývala testováním vytvořeného pluginu a vzorů pro síťové aplikace. Výsledkem testování nově vytvořeného vzoru je zlepšení úspěšnosti detekování protokolu HTTP o *37,94%*. Výsledky testování vzorů základních a nejčastěji používaných síťových protokolů převzatých z *protocolinfo.org* nebyli příliš úspěšné (výsledky jsou uvedeny v tabulce 5.2). Zvýšení úspěšnosti detekce síťových aplikací lze dosáhnout vytvořením kvalitních a aktuálních vzorů. Testováno bylo také zatížení FlowMon sondy při využívání vytvořeného pluginu pro detekci síťových aplikací. Při extrémním provozu nepřesáhlo zatížení sondy *42%*, při běžném provozu poté nepřesáhlo zatížení sondy *16%*.

Kapitola 6

Závěr

Cílem bakalářské práce bylo vytvořit rozšiřující plugin pro sondu FlowMon společnosti INVEA-TECH, který detekuje síťové aplikace a následně informace o detekovaných aplikacích exportuje na kolektor prostřednictvím protokolu IPFIX.

Za tímto účelem jsem vytvořil plugin, který rozpoznává síťové aplikace technologií AppFlow, tedy hloubkovou analýzou paketu, která spočívá v porovnávání datové části paketu se vzory pro síťové aplikace. Pro přesnější informace o rozeznaných aplikacích jsem vytvořil funkce, pomocí kterých lze získat bližší informace o rozeznaných aplikacích z datové části paketu na základě vzorů pro získávání bližších informací. Informace o detekovaných aplikacích jsem připojil k záznamu obsahujícímu základní NetFlow data a vytvořený záznam připravil pro export. Aby byl možný export záznamů obsahujících data o síťových aplikacích, musel jsem vytvořit šablonu pro export a patřičně upravit soubory pro správnou interpretaci dat na straně kolektoru. Export dat prostřednictvím protokolu IPFIX jsem realizoval pomocí existujícího řešení v podobě pluginu *FlowMon IPFIX Export Plugin*[8].

Už z testování během vývoje pluginu vyplynulo, že největší vliv na úspěšnost detekce síťových aplikací mají vzory pro jednotlivé síťové aplikace. Kvůli tomuto zjištění jsem se pokusil vytvořit nový vzor pro protokol HTTP. Kombinace původního a vytvořeného vzoru přineslo zlepšení úspěšnosti detekování protokolu HTTP o 37,94% a to na 95,66%. V závěrečném testování byl plugin testován se vzory pro sedm základních a často používaných síťových protokolů. Také z výsledků závěrečného testování byl patrný velký vliv kvality a aktuálnosti vzorů na úspěšnost detekce aplikací.

Vytvořený plugin může být vítaným přínosem pro administrátory datových sítí. Vzhledem k výsledkům testování by však bylo nejdříve nutné vytvořit kvalitní a aktuální vzory pro rozpoznávání síťových aplikací. Proto bych navrhoval tvorbu kvalitních a aktuálních vzorů jako hlavní směr dalšího vývoje detekce síťových aplikací. Dalším vhodným rozšířením pluginu by bylo implementování algoritmu, který by dokázal předávat k rozeznávání síťových aplikací přednostně vzory těch síťových aplikací, které by byly s největší pravděpodobností očekávány (například na základě použitých portů a transportního protokolu). Tím by se snížilo zatížení FlowMon sondy při použití velkých databází vzorů síťových aplikací.

Literatura

[1] Claise, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. *RFC 5101*, 2008.

[2] Claise, B.; Ed.: Cisco Systems NetFlow Services Export Version 9. *RFC 3954*, 2004.

[3] Elich, M.: *Rozšíření NetFlow kolektoru NfSen o detekci síťových anomálií.* Diplomová práce, Masarykova univerzita, Fakulta informatiky, Brno, 2009.

[4] Fielding, R.; Gettys, J.; Frystyk, H.; aj.: Hypertext Transfer Protocol – HTTP/1.1. *RFC 2616*, 1999.

[5] Oslebo, A.: Application detection using Appflow and passive monitoring. In *Campus network monitoring workshop*, Brno: UNINETT, 2012-04-24.
URL
http://www.ces.net/events/2012/campus-monitoring/p/oslebo-appflow.pdf

[6] Stein, Y. J.; [ipfix-iana at cisco.com]: IP Flow Information Export (IPFIX) Entities. `http://www.iana.org/assignments/ipfix/ipfix.xml`, 2007-05-10, [Last-update 2013-04-29],[cit. 2013-05-1].

[7] Trammell, B.; Boschi, E.; Mark, L.; aj.: Specification of the IP Flow Information Export (IPFIX) File Format. *RFC 5655*, 2009.

[8] Velan, P.: FlowMon - IPFIX Export Plugin. Technická zpráva, Masaryk University, Institute of Computer Science, Brno, 2012.

[9] WWW stránky: The most advanced cloud network platform – citrix. `http://www.citrix.com/products/netscaler-application-delivery-controller/overview.html`, [cit. 2012-12-08].

[10] WWW stránky: Next-generation visibility and real-time control – citrix. `http://www.citrix.com/products/netscaler-application-delivery-controller/features/visibility.html`, [cit. 2012-12-08].

[11] WWW stránky: Lancope's StealthWatch System Now Integrates with Citrix AppFlow. `http://www.lancope.com/blog/lancopes-stealthwatch-system-now-integrates-with-citrix-appflow`, [cit. 2012-12-09].

[12] WWW stránky: Product documentation - Configuring XenDesktop for Load Balancing – citrix.
`http://support.citrix.com/proddocs/topic/netscaler-load-balancing-93/ns-lb-xendesktop-wizard-tsk.html`, [cit. 2012-12-09].

[13] WWW stránky: Free Real-time Appflow Analyzer. http://www.solarwinds.com/products/freetools/appflow-jflow-sflow-analyzer.aspx, [cit. 2012-12-11].

[14] WWW stránky: SolarWinds Knowledge Base. http://knowledgebase.solarwinds.com/kb/images/node%20details%20w%20inactive%20button.png, [cit. 2012-12-11].

[15] WWW stránky: NBAR (Netflow Based Application Detection). http://blogs.manageengine.com/netflowanalyzer/2009/02/17/netflow-based-application-detection-and-qos-implementation-1-of-4/, [cit. 2013-01-10].

[16] WWW stránky: Netflow. http://cs.wikipedia.org/wiki/Netflow, [cit. 2013-01-11].

[17] WWW stránky: Introduction to Cisco IOS NetFlow - A Technical Overview. http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html, [cit. 2013-01-15].

[18] WWW stránky: AppFlow. http://www.appflow.org/, [cit. 2013-01-18].

[19] WWW stránky: FlowMon – INVEA-TECH. http://www.invea.cz/produkty-sluzby/flowmon, [cit. 2013-01-20].

Příloha A

Obsah CD

Přiložené CD obsahuje zdrojové kódy vytvořeného pluginu pro detekci síťových aplikací, databázi vzorů, která obsahuje vzory pro testované síťové protokoly a zdrojové kódy převzatého exportního pluginu určeného pro export záznamů prostřednictvím protokolu IPFIX. Dále obsahuje vytvořenou šablonu pro export a soubory nutné pro správnou interpretaci dat na straně kolektoru. V souboru *readme.txt* je blíže popsán obsah CD a příklad spuštění vytvořeného pluginu na sondě FlowMon. Přiložené CD dále obsahuje zdrojový text této technické zprávy a text technické zprávy ve formátu pdf.

- readme.txt
- l7dec/
  - l7dec.c
  - l7dec-base.c
  - l7dec-base.h
  - export_ipfix.c
  - export_ipfix.h
  - l7dec.config
  - patterns/
  - Makefile
- export_files/
  - ipfix-template-file.txt
  - ipfix-elements.xml
  - fbitdump.xml
- text.pdf
- text/