VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

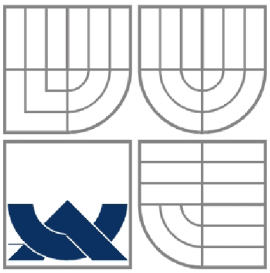
WEBOVÝ PORTÁL S REPORTY O SÍŤOVÉM PROVOZU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

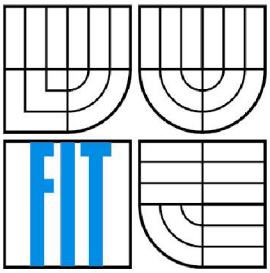
AUTOR PRÁCE
AUTHOR

MILAN DUFEK

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WEBOVÝ PORTÁL S REPORTY O SÍŤOVÉM PROVOZU

WEB PORTAL FOR NETWORK TRAFFIC REPORTING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MILAN DUFEK
AUTHOR

VEDOUCÍ PRÁCE — ING. JIŘÍ TOBOLA
SUPERVISOR

BRNO 2009

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2008/2009

Zadání bakalářské práce

Řešitel: **Dufek Milan**
Obor: Informační technologie
Téma: **Webový portál s reporty o síťovém provozu**
Kategorie: Web

Pokyny:

1. Seznamte se s technologiemi pro tvorbu webových informačních systémů (HTML, CSS, PHP, Javascript, MySQL apod.).
2. Stručně se seznamte s technologií NetFlow pro monitorování sítí.
3. Proveďte analýzu požadavků pro systém umožňující tvorbu reportů, grafů a tabulek na základě NetFlow dat. Systém musí poskytovat podporu široké škály statistik (top uživatelé, nejnavštěvovanější servery, doby činnosti na síti, souhrnné statistiky sítě atp.).
4. Vytvořte detailní návrh tohoto systému a vhodně jej modelujte.
5. Navržený systém realizujte a otestujte, funkčnost systému demonstrujte na vhodně zvoleném vzorku dat.
6. Zhodnoťte dosažené výsledky a diskutujte možnosti dalšího rozšíření systému.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Tobola Jiří, Ing.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2008
Datum odevzdání: 20. května 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Licenční smlouva

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

Abstrakt

Tato práce se zabývá shromažďováním a analýzou statistik o síťovém provozu, jejich uchováváním a zobrazováním. Cílem je jednoduché, přehledné a rychlé uživatelské rozhraní v prostředí webového portálu. Zdrojem statistik je v našem případě technologie NetFlow. Pro dolování nasbíraných dat využíváme nástroj NfDump. Implementace je provedena v jazycích PHP, XHTML, CSS a JavaScript. Pro uložení dat využíváme databázi SQLite.

Klíčová slova

Webový portál, zprávy o síťovém provozu, síťový provoz, síť, databáze, statistiky, NetFlow, NfDump, PHP, XHTML, CSS, JavaScript, SQLite.

Abstract

This thesis deals with gathering and analysis of statistic data about network traffic, it also deals with preserving and displaying such a data. The goal of this thesis is simple and quick user interface in web portal. The source of the statistic data is NetFlow technology. We use NfDump tool for reading the gathered data. Implementation is done in PHP, XHTML, CSS and JavaScript. SQLite Database is used for preserving the data.

Keywords

Web portal, network traffic reports, network traffic, network, database, statistic, NetFlow, NfDump, PHP, XHTML, CSS, JavaScript, SQLite.

Citace

Milan Dufek: Webový portál s reporty o síťovém provoze, bakalářská práce, Brno, FIT VUT v Brně, 2009.

Webový portál s reporty o síťovém provoze

Prohlášení

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

……
Milan Dufek
18.5.2009

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Jiřímu Tobolovi za ochotu a čas strávený při konzultacích této práce. Dále bych rovněž chtěl poděkovat Pavlu Semelovi a Lukáši Petrovickému za cenné vývojářské rady a Haně Žaloudkové za korekturu této zprávy. Osobní poděkování patří také mé rodině za stálou podporu.

Obsah

1 Úvod

V dnešní době, kdy se téměř žádná firma neobejde bez vlastní lokální sítě a připojení na internet, je dobré z důvodu bezpečnosti tyto sítě monitorovat a shromažďovat informace o síťovém provozu. Bezpečnost ale není jediným důvodem k rozhodnutí monitorovat síťový provoz. Důvodů je hned několik, ať se jedná o přehled z hlediska rozložení zátěže na serverech, plánování směrovacích tras, odhalování kolizí v síti, nebo sledování uživatelů v rámci pracovních nařízení. Právě k tomu může sloužit např. technologie NetFlow, která je základem této práce.

Tato technologie se stává čím dál více atraktivní pro menší i větší společnosti. Je to zejména kvůli jejím možnostem stavěným na novějším směru moderní architektury, využívající vlastních hardwareových zařízení pro sběr NetFlow dat. Díky nezávislosti na původnímu vestavění NetFlow exportérů přímo v CISCO směrovačích se stává i výrazně levnějším a otevřenějším řešením. Dále bude řeč o nástrojích pro využití dat. Nástrojů pro zpracování NetFlow dat existuje více, v této práci používám volně šiřitelný a výkonný nástroj NfDump, který byl vyvinut v rámci Evropského projektu GÉANT2, zabývajícím se podporou a výzkumem akademických vysokorychlostních sítí. Podrobněji je technologie NetFlow rozebrána v druhé kapitole.

NetFlow nám nabízí velmi široký potenciál možností. Pro realizaci bakalářské práce je třeba hned na začátku zvolit určité směry, kterými se vydáme. Jedním z těchto směrů je právě vytipování a analýza zajímavých statistik, zejména pro monitorování známých protokolů, portů a nejvytíženějších stanic sítě, dále pak analýza příznaků protokolu TCP a jejich využití k sledování potenciálně napadených počítačů červem nebo pokusů o DoS útok. Tyto statistiky musíme umět zpracovat a následně zobrazovat. Z hlediska dlouhodobějších statistik nám i přes svoji výkonnost NfDump nemůže zaručit chod v reálném čase. V této práci se tedy zabývám právě vývojem aplikace, která umožní získaná data zobrazovat v reálném čase a měnit parametry jejich zobrazení v prostředí webového portálu. Jako prostředník pro uložení již získaných dat nám bude sloužit databáze SQLite, do které budou statistiky automaticky pomocí skriptu ukládány z výstupu NfDumpu. Analýzou a návrhy samotných řešení se zabývám ve třetí a čtvrté kapitole této práce.

V páté kapitole popisuji další technologie, které jsem použité při implementaci této práce, jako jsou XHTML, CSS a PHP pro tvorbu webového dokumentu. Přiblížím práci s databází SQLite, která podporuje databázové transakce a umožňuje tak rychlou a pohodlnou práci s daty. Poté se věnuji stručnému popisu JavaScriptové knihovny Google Visualization API, kterou používám pro tvorbu interaktivních grafů při zobrazování statistik. Na konci kapitoly popisuji zásadní třídy a výkonné skripty výsledné webové aplikace.

Dále následuje popis instalace a závislostí mé práce na systému a softwarovém vybavení serveru, na kterém aplikace bude spuštěna. Samozřejmě také beru ohled na potřeby ze strany klienta, které v práci též nastíním. Již při návrhu jsem zohlednil co nejméně náročné prostředky pro budoucí nasazení aplikace do praxe.

V následujících kapitolách popisuji návrhy dalších řešení a možností zlepšení. V závěru je shrnutí dosažených výsledků a zhodnocení z hlediska dalšího vývoje.

2 Technologie NetFlow

NetFlow se řadí mezi síťové protokoly pro monitorování síťového provozu. Byl vyvinut společností Cisco Systems a původně byl určen jako doplňkový produkt k CISCO směrovačům a přepínačům. Princip by se ve zkratce dal definovat jako měření síťových toků se stejnými vlastnostmi. Právě tyto tzv. IP toky umožňují administrátorům podrobný přehled o síťovém provozu, ať už se jedná o menší lokální, nebo rozsáhlé WAN sítě. NetFlow umožňuje sledování síťového provozu na základě již zmíněných IP toků. Díky nim se dají vyhodnocovat slabá místa sítě, nejaktivnější stanice v síti, DoS (Denial of Service) útoky nebo sledování komunikace kdo-s-kým a délky jejich trvání.

NetFlow protokol vznikl v několika verzích, ale k jeho většímu nasazení došlo až a verzí NetFlow 5, později NetFlow 9, z kterého vzniklo standardizované rozšíření IPFIX (Internet Protocol Flow Information eXport), které se začíná hojně používat v nových směrovačích a přepínačích. Definici požadavků najdeme v RFC3919.

2.1 IP tok

IP tok je základem technologie NetFlow. Tok je v terminologii NetFlow definován jako sekvence paketů se shodnou pětící údajů: cílová/zdrojová IP adresa, cílový/zdrojový port a číslo protokolu. [1] Pro každý započatý IP tok je zaznamenán čas počátku toku, čas konce toku, délka jeho trvání, počet paketů, počet bajtů atd.

Obrázek 1.1: IP tok

2.2 NetFlow architektura

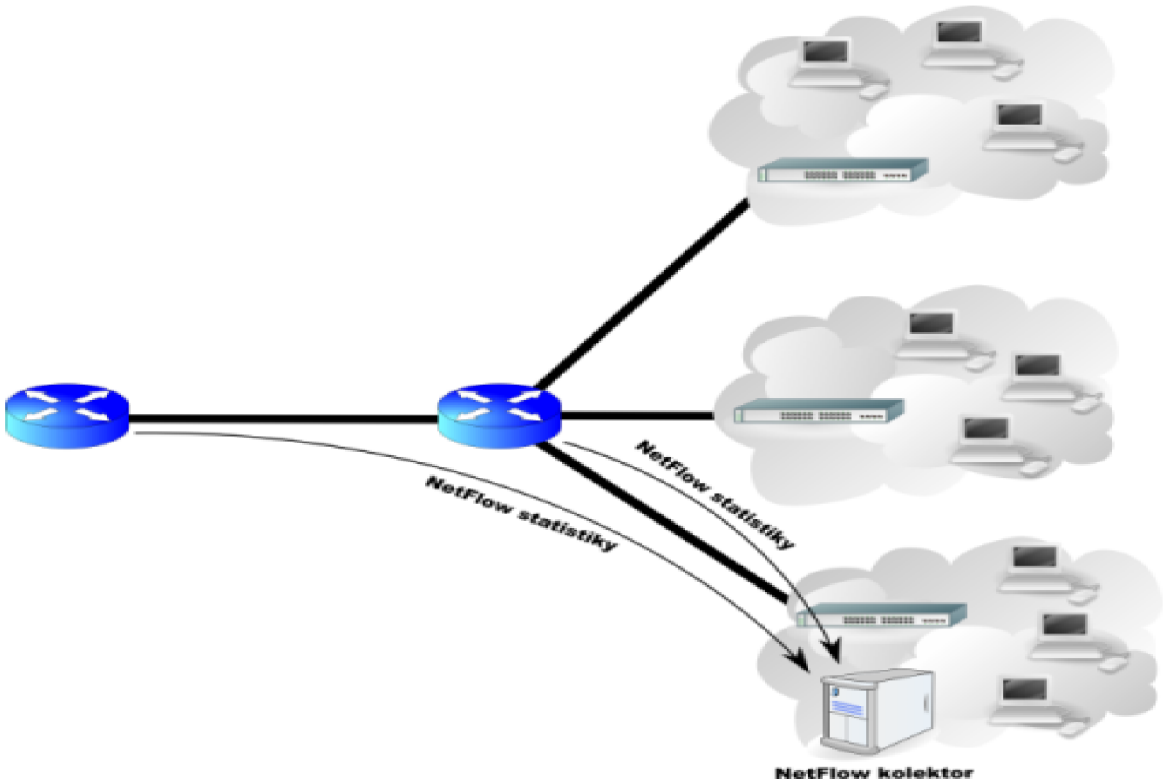
NetFlow architektura se skládá ze dvou základních zařízení: NetFlow exportérů a NetFlow kolektorů. NetFlow exportér je napojen na monitorovaný uzel linky, může být i součástí směrovače. V případě externí sondy se jedná o neviditelné zařízení, tzn. že z hlediska viditelnosti v síti ho nelze identifikovat, se sítí nekomunikuje a chová se jako pasivní prvek. Toto zařízení sbírá zachycené

pakety, ze kterých extrahuje potřebné hodnoty, generuje statistiky a aktualizuje, nebo založí nový záznam o IP toku. Ten poté odesílá na NetFlow kolektor.

NetFlow kolektor je zařízení s velkou úložnou kapacitou, které přijímá záznamy z pravidla z více NetFlow exportérů a ukládá je na diskové zařízení do binárních souborů nebo databáze. Nad uloženými soubory/databázemi zpravidla běží aplikace pro další zpracování a manipulaci těchto dat, v našem případě NfSen. [1, 2]

2.2.1 Tradiční architektura

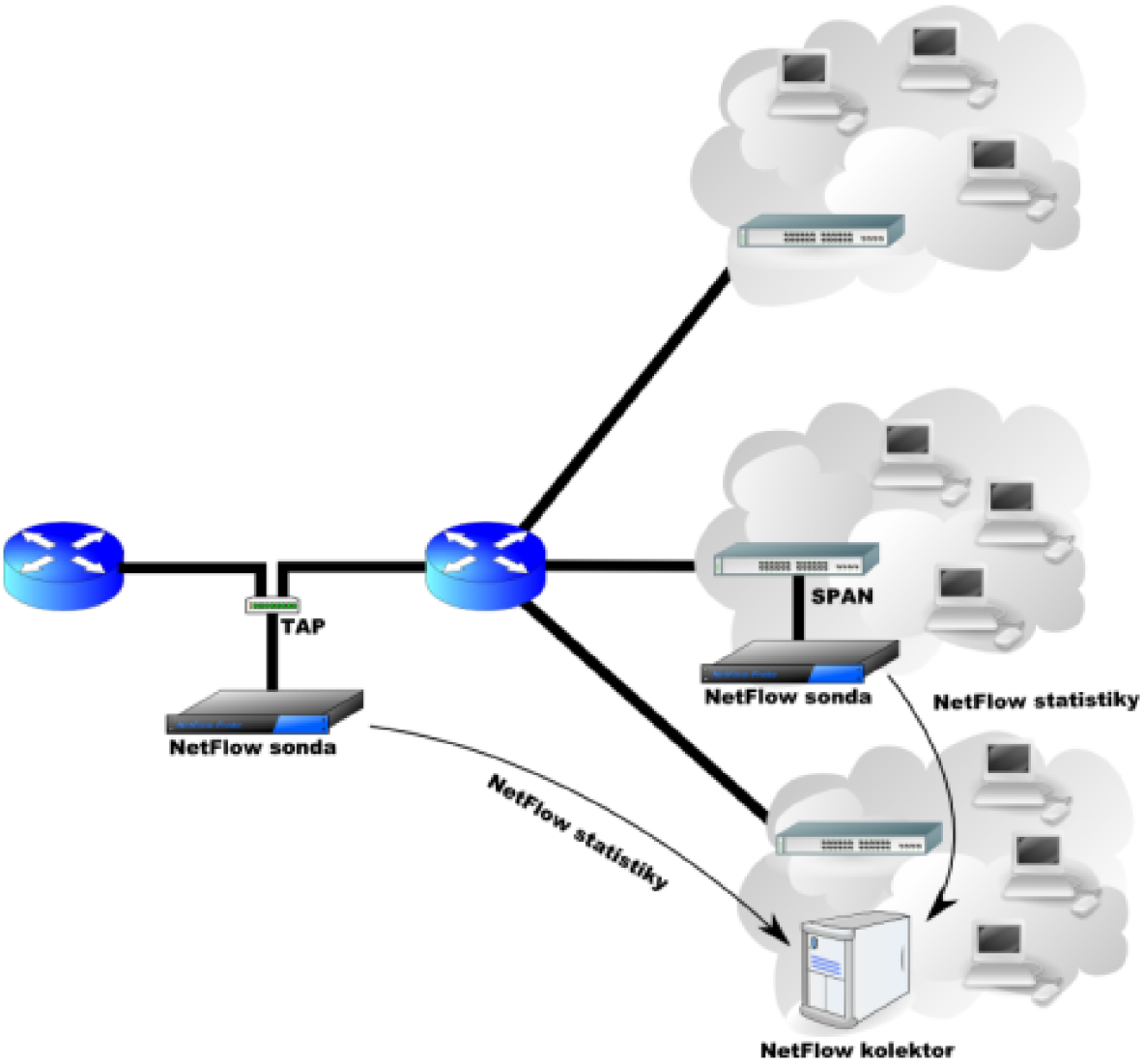
Tradiční architektura navrhnuta firmou CISCO Systems počítá s NetFlow exportérem jako součástí směrovačů, na kterých probíhá celý výpočet NetFlow statistik. To by se mohlo zdát jako výhoda, nepotřebovali bychom další zařízení a každý směrovač by nám současně monitoroval síťový provoz. Ovšem tento přístup má mnohem více negativ. Pořizovací cena takových zařízení bývá poměrně vysoká, čímž trpí především majitelé menších a středních sítí. Ale jako hlavní mínus tohoto řešení je rychlost. Tím, že směrovač musí vypočítávat NetFlow statistiky o síťovém provozu, ubírá na výkonu své hlavní funkce, směrování, což je nežádoucí. Pro částečné zvýšení výkonu takových směrovačů se využívá systém vzorkování IP toků na vstupu, tzn. že se používá pro sběr NetFlow statistik pouze každý N-tý paket, čímž klesá potřebný výkon na sběr těchto statistik. Současně s tím ale klesá i přesnost výsledných statistik, a také pravděpodobnost detekce bezpečnostních incidentů. [1]



Obrázek 2.1: Tradiční NetFlow architektura

2.2.2 Moderní architektura

Z důvodů výše zmíněných nevýhod tradiční architektury začal vznikat nový směr, a to využívání pasivních NetFlow sond (viz. *obrázek 2.2*). Pasivní NetFlow sondy jsou externí zařízení určená pro sběr IP toků, které je dále odesílají na NetFlow kolektor, pokud již není jejich součástí. Cena zařízení je díky jednoduchosti výrazně nižší než u vlastních směrovačů. Takovou sondu je možné připojit do libovolného bodu v síti transparentním způsobem. Jak již jsem zmínil v úvodu, jedná se o pasivní zařízení, které je z hlediska viditelnosti v síti jen velmi těžko zjistitelné, protože pouze odchytává pakety a nijak nezasahuje do vlastních IP toků v síti. Každá taková sonda je napojena na dedikovanou linku, po které odesílá nasbíraná data na NetFlow kolektor. Díky tomu je zajištěna i bezpečnost napadnutí takového zařízení zvenčí a jeho neviditelnost. [1]



Obrázek 2.2: Moderní NetFlow architektura

2.3 Nástroje pro práci s NetFlow

Pro práci s protokolem NetFlow existuje řada nástrojů na dolování znalostí a jejich vizualizaci. Stručně představím některá komerční i volně dostupná řešení.

Jedním směrem jsou aplikace fungující jako softwarové sondy s možností vizualizace dolovaných dat, vytvořené pro využívání NetFlow řešení postavených na CISCO směrovačích a přepínačích. Mezi takové se řadí např. komerční *Flow* Inspektor od Caligare, *NetFlow Analyzer* od Manage Engine, *nGenius* od NetScout nebo *sFlowTrend*™ od inMon. Jejich nevýhodou však může být právě podmíněná vazba na hardware od CISCO Systems, což přináší rizika nekompatibility v případě přechodu na jiný hardware. Naopak jejich výhodou je zaručená podpora ze strany výrobců. V této kategorii existuje ještě více volně dostupných monitorovacích aplikací, např. *NetFlow Monitor* vyvíjený v rámci akademické sítě CESNET, u kterého ale již před lety vývoj skončil. Ze zajímavosti stojí za zmínku nástroj *cflowd* od CAIDA, který se řadí mezi jeden z vůbec prvních nástrojů pro práci s NetFlow daty, s první verzí již v roce 1998. Jeho mladším bratrem je *FlowScan*. Mezi dalšími pouze uvedu např. *fprobe*, *Scrutinizer* pro Windows nebo multifunkční *Stager*. Do této kategorie patří také nástroj *nfdump*, který podrobně představím v následující *kapitole č.2.4*.

Do druhé kategorie patří vlastní hardwareová řešení postavená na moderní NetFlow architektuře. Opomenout bychom určitě neměli také české řešení *FlowMon,* s kterým přišla firma INVEA-TECH, založená členy výzkumného týmu Liberouter ve skupině CESNET.

2.4 NfDump

V předchozí kapitole jsem záměrně nerozebíral volně dostupný nástroj *nfdump*, který jsem si pro práci vybral z důvodu jeho výkonnosti a možnosti volného použití. Tento nástroj byl vyvinut v rámci Evropského projektu GÉANT2, který se zabývá rozšiřováním a podporou akademických vysokorychlostních sítí, a dále také širokým výzkumem, např. v oblasti přenosu obrazu a zvuku, implementací protokolu Ipv6 nebo právě monitorováním síťového provozu vysokorychlostních linek. Z naší země v něm má zastoupení sdružení CESNET.

NfDump je nástroj pro čtení NetFlow záznamu uložených *nfcapd* analyzérem, který ukládá záznamy zpravidla v pětiminutových intervalech do souborů. Adresářová struktura souborů je navržena tak, aby z ní bylo možné data snadno strukturovaně číst voláním nfdumpu např.:

/statistiky_v_miste/protokol/rok/mesic/den/hodina/nfcapd.RRRRMMDDHHmm

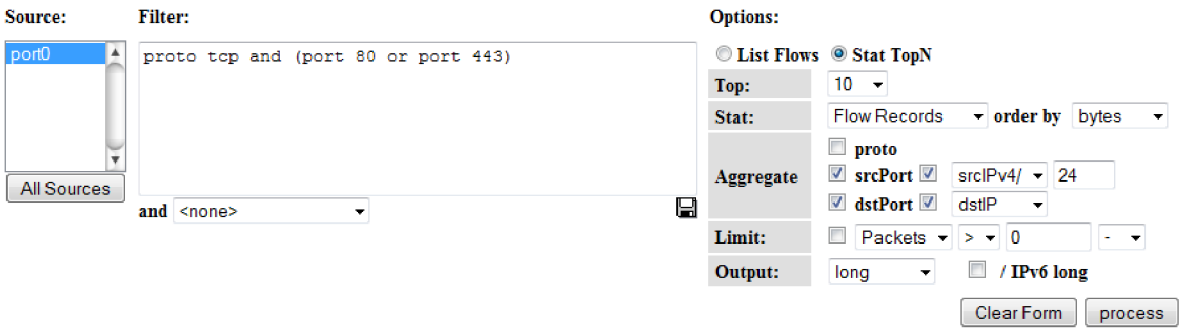
/pobocka_brno/http/2009/04/25/22/nfcapd.200904250000

2.4.1 Součásti nástroje NfDump

- **nfcapd** – je nástroj pro čtení dat ze sítě a jejich ukládání do souborů
- **nfdump** – čte data nashromážděná nástrojem nfcapd, vice v *kapitole 2.3*
- **nfprofile** – čte data nashromážděná pomocí nástroje nfcapd podobně jako nfdump, ale data filtruje podle existujících profilů a uchovává je k pozdějšímu využití
- **nfreplay** – čte nfcapd záznamy a přeposílá je na jiná síťová zařízení
- **nfclean** – script slouží k promazávání starých záznamů
- **ft2nfdump –** převádí *flow-tools* data do formátu srozumitelnému pro nfdump

2.4.2 NfSen

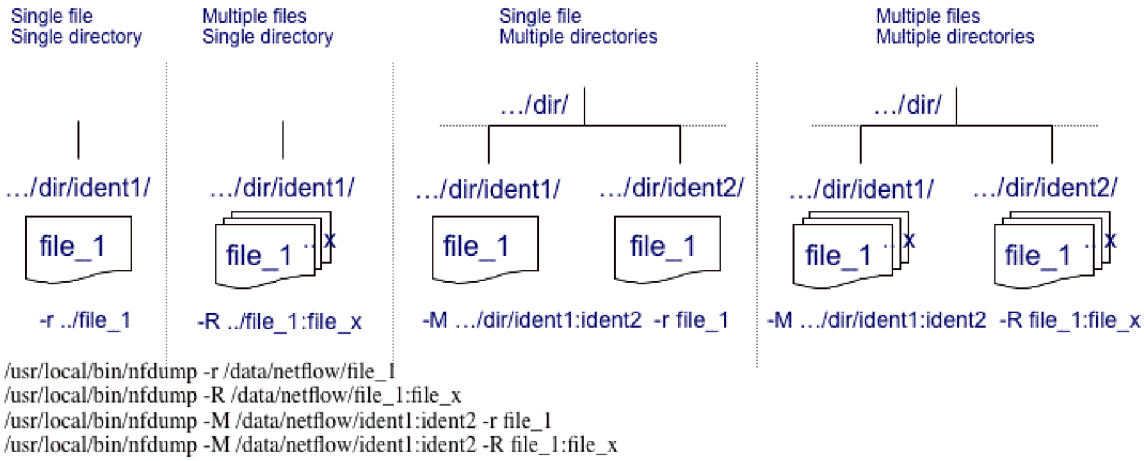
NfSen je grafickou nadstavbou NfDumpu s webovým rozhraním. Umožňuje detailní konfiguraci částí NfDumpu, zmíněných v předchozí kapitole, pomocí přehledného a robustního frontendu. Dále je zde možnost sledovat průběhové statistiky nad nasbíranými NetFlow daty. Je to způsob, který je často využíván správci sítě pro monitorování síťového provozu, ale díky své rozsáhlosti není nejlepším řešením pro běžného uživatele. Ukázka viz. *obrázek č.2.3*.



Obrázek 2.3: Konfigurační formulář nfsenu pro tvorbu nfdump dotazů

2.4.3 Parametry vstupu

Pro ještě snadnější práci se vstupními záznamy slouží parametry nfdumpu **-M** a **-r** resp. **-R**. Přesný význam a praktické využití těchto parametrů popíši na *obrázku 2.4*.



Obrázek 2.4: Struktura nfcapd souborů

Pokud chceme přečíst pouze jeden jediný záznam ze souboru, bude nám stačit parametr -r, který přečte právě ten záznam, ke kterému uvedeme cestu, např. 25.4.2009 v jednu hodinu odpoledne.

nfdump -r /.../http/2009/04/25/13/nfcapd.200904251300

V druhém případě chceme číst záznamy z jednoho adresáře, ale z určitého časového intervalu, např. 25.4.2009 od jedné do šesti hodin odpoledne. Lze číst ale i záznamy z intervalu více dnů. Použijeme parametr **-R**. Pokud tento parametr použijeme bez následného udání intervalu, budou načteny všechny záznamy v zadaném adresáři.

nfdump -R /.../http/.../13/nfcapd.200904251300:/.../18/nfcapd200904251800

V třetím případě budeme chtít číst záznamy z různých protokolů, nebo jiných NetFlow exportérů, přičemž z každého nás zajímá záznam ve stejném čase. Zde použijeme parametr **-M** a **-r**.

nfdump -M /.../http:ftp:ssh/ -r /2009/04/25/13/nfcapd.200904251300

Poslední případ je kombinace dvou předchozích, a tedy čtení záznamu z různých adresářů a ve stejných časových intervalech. Využijeme kombinace parametru **-M** a **-R**.

nfdump -M /.../http:ftp:ssh/
-R /2009/04/25/13/nfcapd.200904251300:/2009/04/25/18/nfcapd.200904252200

Právě toto nám umožňuje nechat si vyvolat záznamy z různých nfcapd úložišť, různých protokolů a časových intervalů. Tak lze velmi snadno měnit rozsah výsledných statistik.

2.4.4 Parametry výstupu

Parametrem **-n** volíme počet řádků výsledných záznamů přečtených nfdumpem. Nejdůležitější je parametr **-s**, kterým volíme typ statistiky/seřazení. Typů statistik je několik: celkové nejobsáhlejší toky *(record)*, unikátní stanice s nejvyšším přenosem *(ip)*, nejvytíženější porty *(port)* nebo nejpožívanější protokoly *(proto)*, atd. Řadit lze podle počtu toků *(flows)*, počtu bajtů *(bytes)* nebo počtu přenesených paketů *(packets)*. Jako příklad uvádím 10 největších toků řazených podle počtu bajtů:

nfdump -M /.../http:ftp:ssh/ -r /2009/04/25/13/nfcapd.200904251300 -n 10 -s record/bytes

Jako další parametr uvádím **-o**, pro určení formátu výstupních dat. Implicitní je formát *long*, který vypíše textovou tabulku klasického typu. Dalším zajímavým parametrem je *raw*, který vypíše pro každý řádek záznamu vlastní tabulku, kde jsou jednotlivé příznaky přiřazeny rovnítkem k hodnotě. Jako poslední uvedu vlastní *custom* formát pro samotné vytvoření výstupu.

Poslední parametr, který uvedu, je **-A** pro agregaci toků podle zdrojových a cílových IP adres a portů. Tento parametr nám umožňuje filtrovat výstup podle podsítí; vlastně slouží jako síťová maska, která omezí rozsah výstupu na daný rozsah adres.

Uvedl jsem zde tedy ty nejdůležitější parametry pro možnosti výstupu, díky kterému můžeme celkem výrazně ovlivňovat typ statistiky, která nás zajímá. [3].

2.4.5 Filtrování výstupu

Parametry výstupu nám již umožnily svým rozsahem pokrýt velkou část výsledných statistik, ale ještě jsou tu filtry, pomocí kterých můžeme filtrovat téměř vše bez omezení. To nám zajistí ještě vyšší specifikaci.

Mezi základní parametry filtru patří např. **proto, ip, port, src, dst, ipv4, ipv6, in, out, flags, duration, pps, bps**, atd. Jak samotné názvy parametrů napovídají, jde o jasné omezení IP adres, portů, protokolu, jejich příznaků apod.

Zápis filtru se provádí do apostrofů. Jako příklad uvedu filtr, který omezí výstup na webové servery, tedy protokol TCP a port 80 nebo 443:

'proto tcp and (port 80 or port 443)'

Filtry jsou opravdu rozsáhlé téma a umožňují nám prakticky cokoliv, více informací naleznete v [3].

2.5 Možnosti využití

Jak jsme si již nastínili, protokol NetFlow má velmi široký potenciál na monitorování síťového provozu a možností využívat znalostí ze záznamů. Poskytuje nám detailní informace o přenosech na nižší úrovni, tedy přímo informace o IP tocích. Na základě IP toků můžeme monitorovat síťovou komunikaci každé stanice, filtrovat spojení dvojice stanic nebo jednotlivých podsítí, dále detekovat útoky či zjistit špatné konfigurace spuštěných služeb. To vše vede k vyšší bezpečnosti počítačové sítě. Záznamy jsou dlouhodobě uchovávány, proto v nich lze snadno dohledávat starší incidenty a zjišťovat detaily o kolizích.

2.5.1 Sledování aplikací

Pomocí IP adres a známých portů je možné detekovat jednotlivé aplikační servery, sledovat jejich provoz a zatížení síťové linky. Pokud bude nějaký server vykazovat vysoký provoz v určité době, lze podle těchto znalostí navrhnout lepší rozložení provozu, konsolidovat více serverů nebo podle velikosti provozu plánovat např. zálohy serverů či jednotlivých stanic.

2.5.2 Sledování uživatelů

S výše zmíněným sledováním aplikací jde ruku v ruce sledování jednotlivých uživatelů. Za předpokladu, že víme, jakou stanici s jakou IP adresou využívá konkrétní uživatel v síti, můžeme odděleně sledovat síťový provoz jeho stanice. Zjistíme tak, které servery navštěvuje a které služby využívá nejčastěji, anebo zda tím neporušuje interní nařízení. Možností je více:

- uživatel s největším přenosem dat
- monitorování zakázaných aplikací
- s kým uživatel komunikuje
- poměr download/upload přenesených dat

2.5.3 Sledování z pohledu poskytovatelů připojení

Pro poskytovatele internetového připojení má NetFlow široké využití. Takový poskytovatel bude mít jistě zájem minimalizovat ceny síťových operaci a současně udržet co nejvyšší výkon. Z NetFlow statistik snadno vyčte potřebné informace o zatížení, rozložení směrování a může optimalizovat cesty směrovačů pro efektivnější a rychlejší provoz. Totožný přístup lze použít i při plánování expanze sítě, anebo výpočtu vhodné agregace. Druhá věc je monitorování uživatelských přenosů a jeho měření z hlediska fakturace - FUP (Fair User Policy). [2]

Dále dodržování vyhlášky č. 485/2005, podle níž je každý provozovatel veřejných komunikačních sítí povinen uchovávat po dobu 3-6ti měsíců údaje o elektronické komunikaci. [4]

2.5.4 Detekce útoků

V dnešní době rozšíření internetu je třeba velmi dbát na bezpečnost počítačové sítě. Můžeme se setkat s útoky různých druhů, ať již se jedná o napadení červem, nebo nějaký z DoS (Denial of Service) útoků. NetFlow dokáže identifikovat rychlý příchod krátkých paketů, pakety bez potvrzení ACK nebo podezřele vysoký provoz na neobvyklém portu.

3 Specifikace požadavků

Úkolem je vytvořit webový portál s reporty o síťovém provozu. To je celkem široký pojem, který je nutno předem specifikovat. Je třeba rozebrat hlavní části, na které musím brát ohled, a podle nich dále navrhnout adekvátní řešení, které splní maximum z našich požadavků.

Základem této práce je technologie NetFlow, která umožňuje podmínky pro velmi širokou škálu různých typů statistik o provozu v síti. Pro čerpání statistik ze zadaného zdroje slouží unixová aplikace NfDump, o níž jsem se již zmínil v předchozí kapitole. Vzhledem k tomu, že cílovou skupinou, které bude moje práce určena, budou spíše lidé v manažerských pozicích než zkušení administrátoři, je třeba vytvořit takové statistiky, které budou pro člověka s nižším IT vzděláním srozumitelné a hlavně zajímavé. Neměly by ale také chybět další statistiky, speciálně ty se zaměřením na bezpečnost síťového provozu, sledování různých anomálií v provozu nebo detekci DoS útoků.

Jak již bylo řečeno jedná se o webový portál, který má umožnit širší škálu různých reportů o síťovém provozu a jejich zobrazování prostřednictvím webu. K zobrazení naší aplikace je tedy třeba tvorba webového dokumentu HTML, resp. XHTML pro tvorbu vzhledu kaskádových stylů CSS, případně pro tzv. oživení aplikace využití JavaScriptů. Samotné generování kódu dokumentu na serveru bude probíhat s využitím nějakého skriptovacího jazyka pro generování HTML, resp. XHTML kódu. Z hlediska jednoduchosti, možnosti širšího nasazení a budoucí rozšiřitelnosti jsem zvolil jazyk PHP, který se pro tento účel nabízí.

Výsledkem bude webová aplikace s přehledným, rychlým a jednoduchým ovládáním, umožnující snadno měnit druh statistik a jejich období. Co se týče vzhledu, tak by rozhraní mělo být i atraktivní, aby se v něm uživatel cítil příjemně a měl vše intuitivně při ruce, snadno se orientoval a nemusel dlouze hledat potřebný záznam nebo způsob jakým ho zobrazit.

V neposlední řadě automaticky řešit získávání dat na pozadí aplikace, tedy čerpání dat pomocí volně dostupného nástroje NfDump, který jsem pro tuto práci zvolil, a umožňovat snadné rozšíření o další statistiky dle potřeby. Aplikace by měla být použitelná v praxi a otevřená dalším rozšířením. Získávání statistik by mělo být automatické, nejlépe spouštěné v pravidelných intervalech ve správně vytipovaném čase, aby docházelo k co nejmenšímu zatížení na serveru, a ve vhodnou dobu, kdy server není zatížený jinými úkony.

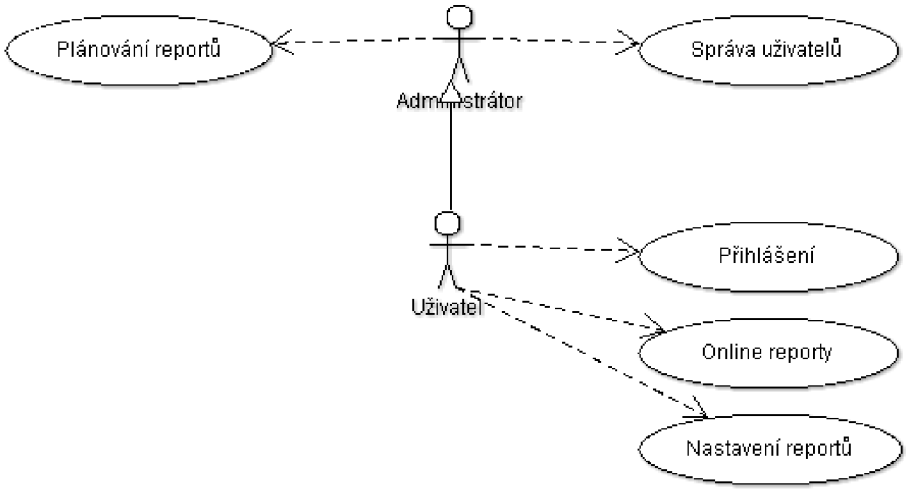
Jelikož se jedná o webové rozhraní, které může být přístupné zvenčí, je třeba zajistit zabezpečení pomocí autentizace oprávněných uživatel a podporu více skupin, např. uživatelů, kteří mohou pouze prohlížet statistiky, nebo administrátorů, co mohou jejich účty spravovat. Každý uživatel by měl mít možnost vytvářet si vlastní pořadí statistik a smět ho měnit.

4 Analýza a návrh řešení

Po specifikaci zadání se podrobně podívám na již zmíněné faktory ovlivňující návrh řešení. Rozeberu, s jakými nápady jsem přišel, a problémy, které bylo třeba řešit. V návrhu jsem zohledňoval několik hlavních bodů zadání, s kterými jsem musel počítat již na začátku. Především byl kladen důraz na širokou škálu různých statistik a jejich zajímavost, ale také i na tvorbu jednoduchého a intuitivního uživatelského rozhraní, celkovou rychlost aplikace a její bezpečnost. Dalším důležitým kritériem bylo také umožnění automatického běhu bez dalších zásahů uživatele po jejím spuštění na serveru tak, aby byla vždy zaručena aktuálnost zobrazovaných dat.

4.1 Diagram případů užití

Diagram případů užití, vytvořený v jazyce UML (Unifield Modelling Language), je určen k definici chování systému z pohledu uživatele. Zobrazuje role uživatelů v systému a k nim přiřazuje jednotlivé akce, které mohou provádět. Diagram případů užití vytvořený během této práci je na *obrázku č.4.1*.



Obrázek 4.1: Diagram případů užití

4.2 Typy statistik

NetFlow umožňuje vyhodnocení opravdu velmi široké škály statistik, což je způsobeno právě sledováním IP toků. Jen samotný NfDump nabízí přes deset typu základních filtrů, v kombinaci dalších možností agregace a řazení roste počet možností exponenciálně. Ovšem většina těchto statistik by pro běžného manažera nebyla natolik zajímavá.

Administrátora, který plánuje např. směrovací trasy jistě velmi zaujme statistika průměrného vytížení jednotlivých podsítí v rámci jemu uděleném adresovém prostoru. Avšak pro manažera by byla tato statistika méně zajímavá.

Právě proto jsem se zaměřil na vyhodnocení a předpřipravení statistik obecnějšího charakteru a z hlediska experimentu jsem zařadil možnosti monitorování špatné konfigurace a detekci DoS útoků různých typů, podrobnější rozbor je další *kapitole č.4.1.2*. Mezi statistikami najdeme např.:

- stanice s nejvyšším počtem přenesených dat do internetu/místní sítě
- nejvyšší objem přenesených dat na protokolech http, ssh, ftp apod.
- nejnavštěvovanější servery
- detekce DoS útoků

4.3 Detekce DoS útoků

Zvláštní kapitolou je monitorování DoS útoků v síti. Jedná se o rozsáhlou kapitolu, v níž jsem se snažil zaměřit na monitorovatelné útoky nad protokolem TCP. DoS útoků může být více druhů a mohou být distribuované, tedy prováděné více než jednou stanicí. Jejich hlavním cílem je zahltit cílovou stanici nebo server takovým množstvím požadavků tak, aby zablokovaly celou šířku jeho přípojné linky nebo vyčerpaly jeho systémové prostředky, což může vést k zhroucení systému a jeho dočasné vyřazení z provozu.

Protokol TCP má celkem 6 základních příznaků:

- **URG** – značí důležitá data
- **ACK** – potvrzuje správně přenesené pakety
- **PSH** – paket nesoucí aplikační data
- **RST** – odmítnutí spojení
- **SYN** – navázání spojení
- **FIN** – dokončení navázaného spojení

Jedním z útoků je tzv. *SYN flood,* jenž může být detekován pomocí filtrování paketů na TCP spojení, u kterých byl odeslán paket s příznakem SYN bez následujícího ACK. Server odpoví klientovi paketem s příznaky SYN+ACK, ale ACK nedorazí. To v praxi vede k tomu, že zasláním více takovýchto paketů alokuje veškeré systémové prostředky na vytvoření nových spojení, která jsou uzavřena pouze napůl a nedovolují uvolnit linku pro nová. Téměř analogicky pracuje tzv. *FIN flood*, který po ustanoveném spojení nezašle ukončující paket obsahující příznak FIN.

Dalším potenciálním narušením je, že se nějaký klient snaží skenovat naše porty a zjišťovat tak informace o tom, zda jsou otevřené či nikoliv.

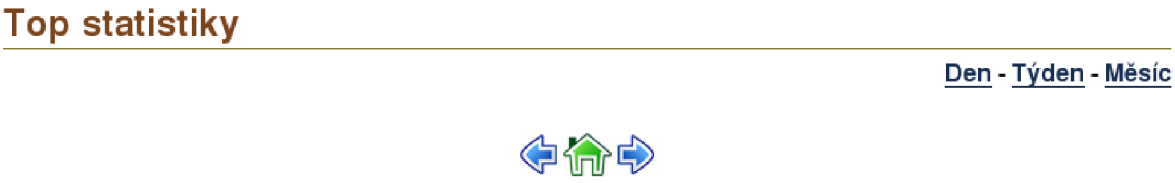
Prvním je *Xmas-scan*, kdy klient odešle na server paket se všemi příznaky, někdy se uvádí pouze příznaky URG, PSH a FIN. To značí nestandardní nastavení a server si vyžaduje potvrzení od klienta. Opakem Xmas-scanu je *Null scan*, který docílí podobného efektu zasláním paketu bez příznaků, čímž je vrácen RST ze strany serveru.

Tato nastavení paketů jdou pomocí NetFlow snadno zaznamenat a lze z nich vyvodit další opatření. [21]

4.4 Uživatelské rozhraní

V dnešní době pokročilých technologií a při velké konkurence schopnosti se zvláště při tvorbě internetových prezentací dbá na atraktivní vzhled aplikace a snadné, intuitivní ovládání. Tento bod proto patří k důležitým.

Cílem této práce však vzhledem k její povaze a rozsahu samozřejmě není vytvořit profesionální grafický návrh výsledného portálu. Důraz proto kladu na intuitivní prvky při ovládání, aby se uživatel v prostředí cítil dobře a aby mu umožnilo jednoduchými operacemi měnit intervaly a rozsahy statistik.



Obrázek 4.2: Panel pro změny intervalů

4.5 Rychlost

Aplikace by měla reagovat rychle v reálném čase. Volání libovolného NfDump dotazu v intervalu jednoho měsíce na malé lokální síti do deseti počítačů, trvá běžně okolo 2-3 vteřin. Na větších

lokálních sítích nebo WAN sítích délka čekání na výsledky dotazu hodinových intervalu může trvat řádově v minutách. Aplikace by musela podobný dotaz volat s rozdílnými parametry několikrát na jedné obrazovce. Dolování dat pomocí NfDumpu a následné zobrazování v reálném čase je tedy pro tento účel nevyhovující. A to jsem ještě nezmínil, jak by toto neustálé dotazování zatěžovalo server a nežádoucím způsobem zpomalovalo celý proces.

Řešením může být například dané dotazy vzorkovat a ukládat pouze výsledky dotazů pro různé typy statistik. Možností jsem měl několik. Ukládat výsledky do souboru, pro každý výsledek nový soubor – to by nebylo moc dobré řešení při stovkách dotazů týdně. Ukládat například po dnech by zase zvyšovalo složitost algoritmu na procházení souborů, což by se sice dalo snadno vyřešit pomocí XML struktury dokumentu, ale jako lepší možnost jsem nakonec vyhodnotil použití databáze.

Databáze je pro větší množství dat stavěná a práce s daty se díky SQL podstatně zjednoduší. Zbývalo už jen vybrat databázový server. To přináší komplikace z hlediska instalace na serveru, i když ve většině firem nějaký ten databázový server běží, nemusí to však být pravidlem. Vybral jsem proto, trochu i z experimentálních důvodů, odlehčenou formu databáze SQLite. Ta využívá databázového souboru a nevyžaduje spuštěný server. V PHP je pro pohodlnou práci s SQLite knihovna PDO (PHP Data Objects) podporující základní operace nad databází včetně transakcí, které jsou nutné pro udržení vzájemné konzistence tabulek databáze během aktualizací.

4.6 Aktuálnost

Aktuálnost dat byla jednou z klíčových potřeb. Jak jsem již zmínil v předchozím bodě, data získaná z výstupu NfDumpu se uloží do databáze, z ní se pak kdykoliv načtou a zobrazí. Jedna část aplikace se tedy stará o zobrazování existujících statistik a druhá o jejich pravidelné získávání pomocí NfDumpu. K tomuto získávání jsem zvolil naplánování pravidelného spouštění výkonného kódu v unixové aplikaci *Cron*.

Cron démon je unixová aplikace pro plánování úloh, převážně využívaná pro automatické spouštění různých skriptů v příkazové řádce. Využívá tzv. cronetables, což jsou tabulky, v kterých jsou zaznamenány naplánované úlohy. Ve Windows se nachází obdobná aplikace *Plánovač úloh*. Cron nám tedy zajistí pravidelné spouštění výkonného skriptu **nfdump.php** pro vzorkování vybraných statistik voláním dotazů aplikace NfDump a ukládaní výsledků do databáze. Naplánování událostí můžeme udělat přímo na serveru v příkazové řádce nebo vzdáleně přes zabezpečené SSH připojení. Z Windows lze použít např. aplikaci Putty. Seznam naplánovaných úloh zobrazíme příkazem **cronetab -l** a editovat jej můžeme příkazem **cronetab -e**. Záznam v cronetab vypadá následovně:

*** * * * * aplikace /cesta/ke/skriptu**

Hvězdičky představují čas spouštění, od začátku: minuta, hodina, den v měsíci, měsíc a den v týdnu. Poté následuje aplikace a za ní skript, který má spouštět. V našem případě tedy budeme spouštět skript nfdump.php aplikací php, každý den v 1:00. Tento čas jsem zvolil jako vhodný jednak kvůli aktuálnosti statistik z předchozího, jednak kvůli nízké vytíženosti serveru v tuto hodinu.

0 1 * * * php /var/www/nfdump.php

4.7 Dostupnost

Dalším hlediskem je dostupnost aplikace: díky charakteru webového rozhraní se počítá s dostupností aplikace z internetu. Každý uživatel bude moci prohlížet statistiky z libovolného místa; jedinými předpoklady je připojení k internetu a webový prohlížeč, podporující JavaScript, což je v dnešní době téměř každý. Podporovány budou prohlížeče: FireFox, Internet Explorer 7+, Opera a Safari.

Při dostupnosti bude jistě vhodné zajistit ukládání uživatelských nastavení na straně serveru místo často používaných cookies, které uživatel s přechodem na jiný počítač již nemá nastavené. Ideální bude mít pro každého uživatele vlastní záznam v databázové tabulce. Tím bude zajištěna dostupnost aplikace se stejným nastavením bez omezení na libovolném počítači.

4.8 Bezpečnost

S předchozím bodem souvisí bezpečnost aplikace. Jelikož bude aplikace dostupná z internetu, je třeba zajistit přihlašování do systému. Pro zajištění bezpečnosti uživatelských dat budu využívat kryptovacího algoritmu SHA1 (Secure Hash Algorithm 1). Jedná se o hashovací funkci vytvářející 160-ti bitový otisk obrazu o délce 40-ti znaků. SHA1 je jakýsi pokračovatel funkce MD5, která vytváří 32 znaky dlouhé otisky. V mojí aplikaci budu tuto funkci používat především na uložení a posílání citlivých dat mezi klientem a serverem, jako jsou např. hesla uživatelů.

5 Implementace

Když byl hotový návrh mé aplikace, soustředil jsem se na implementaci. V této kapitole charakterizuji jazyky a knihovny použité při implementaci. Na závěr popíši hlavní implementované třídy a skripty.

5.1 Použité technologie

Na začátku bylo třeba zvolit technologie pro tvorbu výsledné aplikace. Naše aplikace spadá do kategorie web, proto jsem zvolil kvůli přísnější struktuře dokumentu značkovací jazyk XHTML 1.0 Strict, což umožní širší podporu internetových prohlížečů. Dále jsem pak zvolil kaskádové styly CSS pro tvorbu vzhledu a formátování textu a pro zabezpečení prvků dokumentu na straně klienta je použit skriptovací jazyk JavaScript. S ohledem na mé zkušenosti a charakter aplikace jsem při výběru skriptovacího jazyka volil PHP, pro relativně snadnou implementaci a výbornou dokumentaci, a HTTP server Apache pro jeho výkonnost. Jako úložiště dat jsem zvolil databázi SQLite.

5.1.1 XHTML

XHTML (Extensible Hypertext Markup Language) je značkovací jazyk pro tvorbu hypertextových dokumentů ve WWW prostředí vyvíjený konsorciem W3C. Jedná se o rozšíření HTML (Hypertext Markup Language) tak, aby vyhovoval podmínkám XML (Extensible Markup Langueage) dokumentů a přitom byla zachována zpětná kompatibilita. XHTML je definován ve třech verzích: *Strict*, *Transitional* a *Frameset* podle přísnosti psaní dokumentu a míry zpětné kompatibility s HTML 4.1.

Hlavní rozdíly oproti HTML jsou přísná pravidla pro dodržování XML struktury, především správné zanořování jednotlivých elementů ve struktuře dokumentu. Každý párový element musí být uzavřený. Pokud není párový, jako např. *
* volí se zápis se zpětným lomítkem před uzavírací závorkou *
*. Názvy jednotlivých elementů musí být psány malými písmeny, a to je způsobeno case sensitive vlastností XML. Dále pak všechny atributy elementů, včetně číselných, musí být v uvozovkách. Některé zastaralé elementy a atributy fyzického formátování byly zakázány, jako např. iframe, font, s nebo center. Tyto elementy plně nahrazuje podpora kaskádových stylů CSS. [5]

5.1.2 CSS

CSS (Cascading Style Sheets) je jazyk vyvinutý konsorciem W3C. Byl navržen za účelem oddělit vzhled od struktury a obsahu dokumentu HTML, XHTML a XML. To umožňuje snadnou změnu vzhledu bez zásahu do struktury dokumentu, tvorbu více vzhledů pro jeden dokument nebo tiskové sestavy. Nevýhodou je jeho podpora a implementace v prohlížečích, především starší prohlížeče, jako např. Internet Explorer 6, zobrazují některé elementy odlišně od jiných prohlížečů. Tento problém naštěstí již pomalu zaniká s přechodem na vyšší verze prohlížečů. V současnosti se pracuje již na CSS třetí generace. [5]

5.1.3 PHP

PHP (Hypertext preprocesor, původně Personal Home Page) je skriptovací jazyk zejména pro generování dynamických webových stránek, ale lze ho použít i pro tvorbu konzolových a desktop aplikací. Jazyk není závislý na konkrétní platformě a jeho syntaxe vychází z jazyků Perl, C a Java.

PHP skripty lze přímo kombinovat v těle dokumentu s HTML resp. XHTML kódem, což umožňuje velkou volnost a kreativitu při práci, ale je třeba se držet určitých pravidel, aby byla zachována přehlednost kódu. Jazyk je velmi populární především díky relativně snadné implementaci, široké podpoře ze strany veřejných poskytovatelů a obsáhlé dokumentaci.

Nyní jazyk PHP ve verzi 5 má plně objektovou podporu, která otevřela nový směr s vyšší abstrakcí a umožnila na PHP stavět robustnější a přehlednější aplikace se snadnější možností budoucích rozšíření. Je na něm založeno mnoho frameworků, z nejznámějších např. Zend, Kohana, Nette nebo CakePHP. Pro PHP existuje i velká řada knihoven pro práci s XML databázemi jako MySQL, PostgreSQL, SQLite nebo Oracle, PHPmailer, exporty do formátů PDF atd. [6]

5.1.4 JavaScript

JavaScript je multiplatformní objektově orientovaný jazyk spouštěný na straně klienta po načtení skriptu z těla dokumentu. Zpravidla je využíván ve webových aplikacích pro dynamickou práci s prvky webové stránky bez nutného opětovného načítání, například pro chybové hlášení nebo zobrazení/skrytí části dokumentu. Nelze na něho ale vždy spoléhat, protože nemusí být v prohlížeči implementován nebo může být vypnut. [9]

5.1.5 SQLite

SQLite je jednoduchý multiplatformní relační databázový systém. Nevyžaduje spuštěný vlastní server a ukládá databázi do strukturovaného souboru pomocí hashovacích technik s primárním klíčem. Má relativně bohatou podporu ze strany různých programovacích a skriptovacích jazyků v podobě

knihoven. Podporuje základní datové typy, indexy a základní SQL syntaxi. Pro zajímavost je využívána například jádrem webového prohlížeče FireFox.

5.2 Použité knihovny

5.2.1 Google Visualization API

Google Visualization API, dále jen GAPI, je rozsáhlá open source knihovna napsaná v jazyce JavaScript a AJAX. Umožňuje především práci s grafikou v prostředí webu, jako jsou např. grafy, mapy, různá schémata apod. Využití GAPI je pravdu rozsáhlé; v této práci používám tuto knihovnu pro kreslení koláčových grafů, která je pouze jednou z jeho částí. Ukázku reportu najdeme na *obrázku č.5.1*.

5.3 Popis hlavních skriptů

Tématem této kapitoly je samotná implementaci PHP skriptů. Popíšu v ní některé důležitější třídy a skripty s výkonným kódem, který je nezbytný pro funkci celé aplikace. Popisování použitých algoritmů se nebudu věnovat do hloubky, protože velmi dobrý komentář a podrobnější informace přímo o PHP skriptech najdeme v příručce k programu, která se nachází v *Příloze č.1*.

index.php – Hlavní skript každé internetové aplikace, který se načítá jako první v kořenu adresářové struktury webu. Obsahuje veškeré informace o typu dokumentu, hlavičku dokumentu, odkazy na kaskádové styly CSS, hlavičky JavaScriptové knihovny **jsAPI**. Dále ještě připojuje konfigurační soubor **config.inc.php**, který popisuji v dalším odstavci. Index se dále stará o vytváření session a bufferování výstupu dokumentu. Kontroluje se, zda je uživatel přihlášen a jaká má uživatelská práva. V závislosti na jeho právech, se zde vytváří menu aplikace. Administrátor vidí více položek než běžný uživatel. Poté probíhá připojování dalších řídících skriptů dle volby.

config.inc.php – Konfigurační soubor důležitý zejména pro načtení všech ostatních tříd odděleně od výkonného kódu. Vytvoření instance, spojení s databází a definování přístupových práv jednotlivých stránek aplikace.

NfDump.class.php – První ze dvou zásadních tříd celé aplikace starající se o operace na pozadí, tedy tzv. backend aplikace. Obsahuje metody pro sestavení dotazu NfDumpu, metodu pro jeho volání, zpracování získaných dat a jejich uložení do SQLite databáze v korektním formátu.

nfdump-auto.php – Výkonný kód backendu aplikace slouží především pro vytváření instancí NfDump.class za účelem uložení zpracovaných dat do databáze. Je speciálně navržen pro automatické volání pomocí aplikace Cron, viz. *kapitola 4.1.4.* Při volání metod využívám

databázových transakcí. Pro zrychlení chodu, menší zatížení serveru a udržení konzistence vazeb mezi tabulkami.

nfdump-manual.php – Obdobný skript jako nfdump-auto.php, ovšem uzpůsobený na jednorázové ruční použití. Speciálně pro prvotní vytvoření vzorků zpětně v čase, tedy uložení dat za předchozí časové období, kdy ještě moje aplikace nebyla na serveru využívána.

TopStats.class.php – Jak již bylo předesláno, druhá ze zásadních částí aplikace, starající se o zobrazování již zpracovaných statistik, tedy tzv. frontend aplikace. Obsahuje metody pro načtení předpřipravených statistik z databáze, jejich vypsání do příslušných tabulek a tvorbu grafů pomocí GAPI. Dále několik pomocných metod pro zpracování.

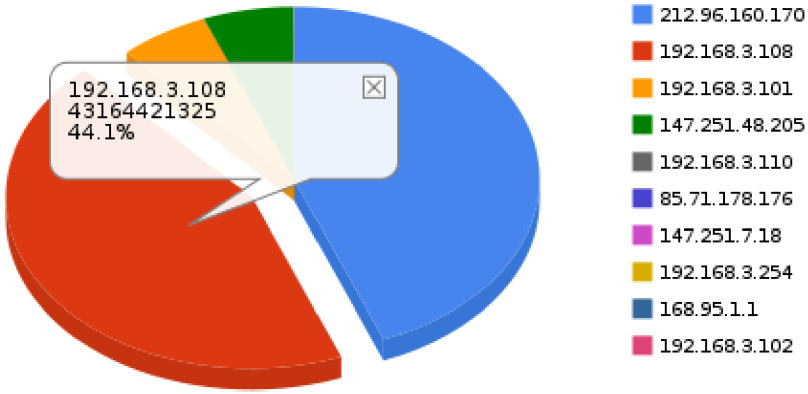
topstats.php – Výkonný kód frontendu aplikace, pro vytváření instancí jednotlivých statistik a souhrnných přehledů. V neposlední řadě se tu volá instance vykreslující řídící menu pro změny period a intervalů zobrazovaných statistik.

Registration.class.php – Třída obsahující metody pro správu uživatelských účtu, tedy např. vytvoření nového účtu, mazání apod.

Login.class.php – Tato třída implementuje metody pro přihlašování uživatelů, jejich odhlašování, metody pro kontroly oprávnění a nebo tisk přihlašovacího formuláře.

Math.class.php – Třída se statickými metodami pro převody binárních jednotek, hojně využívaných v předchozích třídách.

Nejvytíženější stanice (Sobota 28.03.2009)



#	IP adresa	Počet paketů	Data
1.	212.96.160.170	42467328	40.2 GB
2.	192.168.3.108	42362470	40.2 GB
3.	192.168.3.101	5452595	5.4 GB
4.	147.251.48.205	5347738	5.4 GB
5.	192.168.3.110	381193	39.1 MB
6.	85.71.178.176	31961	9 MB
7.	147.251.7.18	71741	7 MB
8.	192.168.3.254	46608	4.3 MB
9.	168.95.1.1	47274	3.9 MB
10.	192.168.3.102	37699	2.6 MB

Obrázek 5.1: Ukázka online reportu

Souhrnné statistiky za období

Týden od: 16.03.2009 **do:** 22.03.2009

Celkový počet toků:	1402694	**Průměrný počet bajtů v toku:**	986
Celkový počet bajtů:	289.8 GB	**Průměrný počet bajtů:**	3.9 MB
Celkový počet paketů:	315306803	**Průměrný počet paketů:**	521 B

Obrázek 5.2: Ukázka souhrnných statistik

6 Instalace a testování

Samotnou implementací vývoj aplikace ještě nekončí. Aby mohla být aplikace předvedena koncovému uživateli, je třeba ji řádně otestovat a odhalit co nejvíce chyb. Některé chyby se vždy vyskytnou a je třeba je odhalovat co nejdříve, ideálně je zachytit již během vývoje. V této kapitole popisuji závislosti na operačním systému a dalším softwarovém vybavení. Popis vlastní instalace přímo na server zde popisovat nebudu, je uveden v příloze README.txt. Moje práce byla vyvíjena a testována na následujícím vybavení:

Server:

- OS: Linux Ubuntu: Kernel 2.6.27-11 – generic
- NfDump 1.5.7
- Apache 2.0
- PHP verze 5.2.6-2
- PDO SQLite 3.5.9

Klient:

- OS: Linux Ubuntu: Kernel 2.6.27-11 – generic, Windows XP
- Prohlížeče: FireFox 3.0, Opera 9.6, Internet Explorer 7

6.1 Požadavky

Specifikace minimálních požadavků na straně serveru a klienta, aby bylo možné aplikaci používat bez omezení.

6.1.1 Na straně serveru

- Operační systém unixového typu, např. Linux nebo FreeBSD
- Nainstalovaný NfDump 1.5.x a přístup k nfcapd záznamům
- Spuštěný HTTP server např. Apache
- Nainstalované PHP 5.x
- Doinstalovaná podpora PDO SQLite 3.x pro PHP

6.1.2 Na straně klienta

Zde stačí libovolný operační systém s připojením k internetu a webový prohlížeč s povoleným JavaScriptem, např. FireFox 2+, Internet Explorer 7+ nebo Opera.

7 Možná rozšíření

Vylepšení jsou vždy možná, a zvláště pokud se má práce pochybuje v nepříliš probádané oblasti jako je využívání technologie NetFlow pro širší monitorování síťového provozu. Navrhuji možnosti dalších směrů, kterými by se mohla má práce dále rozšířit, ale již pro to nebyl prostor v rámci bakalářské práce.

7.1 Rozšíření statistik

Technologie NetFlow a nástroj NfDump nabízí velmi širokou oblast statistik a použití. Podle specifikace je tato práce určena spíše pro manažery než správce sítě, ale charakter by se mohl více rozšířit. Dále pak by se mohlo přidat více statistik, a to i na více specifické oblasti jako je sledování směrování nebo rozložení zátěže. Ve své práci využívám především filtrování IP adres, portů a protokolů, což je pouze část možného rozsahu. Další je možnost zahrnutí souhrnných reportů za časové intervaly, tzv. traffic reporty, s průběhovými grafy. Ty by umožnily širší pohled na síť a usnadnily monitorovat např. takové věci, jako největší aktivitu serverů během dnů. S tím je spojena i další možnost detekce DoS útoků. To je tak rozsáhlá kapitola, že by sama o sobě vystačila na téma bakalářské práce.

7.2 Bezpečnost

Zabezpečení aplikace je další kategorie, kde je neustále co vylepšovat; co bylo dnes postačující nemusí být již zítra pravda. Mezi možné kroky při zlepšování bezpečnosti lze zahrnout použití zabezpečeného internetového protokolu HTTPS, který využívá šifrovaného přenosu pomocí protokolů SSL nebo TSL. Lze tak zmenšit riziko odposlouchávání, podvržení přenášených dat z třetí strany, a také ověřovat identitu protistrany.

7.3 Přímá administrace reportů

V mé práci je jednoduché administrační rozhraní umožňující pouze základní operace s uživatelskými účty, jako je jejich vytváření nebo odstraňování. Rozšíření administračního rozhraní je možné i na jiné oblasti aplikace, např. vytváření nových typů statistik přímo přes webové rozhraní a jejich spravování.

Pro případ, že uživatel nechce sledovat a využívat webového rozhraní, by mohla být užitečnou také možnost exportování dokumentu do jiných offline použitelných formátů. Tedy možnost

exportovat dokument např. do PDF. Uživatel by měl možnost si zvolit mezi prohlížením interaktivního webu, nebo statickým dokumentem, který by si ale mohl nechat pravidelně zasílat přes email v zvoleném čase.

7.4 Uživatelské rozhraní

Webdesign je další z kapitol, kde jde vývoj stále dopředu a kde velmi rychle přichází nové trendy a technologie. S příchodem něčeho nového odchází to staré. Dnešním trendem je např. AJAX. Jistě by se z jeho potenciálu dalo čerpat i v mé práci a oživit tak další prvky portálu pro usnadnění ovládání a vyšší interaktivitu. Ve stávající podobě je portál navrhnut v celku jednoduše ale již v návrhu je počítáno s dalším rozšířením uživatelského rozhraní do budoucna. Vzhled není doplněn profesionální grafikou, ale pouze kaskádovými styly CSS; i zde je možnost zlepšení k větší atraktivitě a přehlednosti. Případně by se mohla přidat i možnost volby z více vzhledů.

8 Závěr

Na základě znalostí technologie NetFlow a seznámení se s dalšími technologiemi k tomu potřebnými, byla provedena analýza potřeb se zaměřením na atraktivitu a použitelnost statistik o síťovém provozu. Podařilo se mi vymyslet jak souhrnné statistiky se zaměřením spíše obecného charakteru, jako např. nejaktivnější stanice, nejnavštěvovanější servery apod., tak detekce potenciálních útočníků při skenování portů přes protokol TCP s různými typy příznaků nebo detekce SYN/FIN záplavového DoS útoku. Další věcí je navržené automatické vzorkování nastavených typů statistik do SQLite databáze; výběr této databáze se ukázal jako dobrá volba a jako zajímavá alternativa k populární MySQL. Díky využití databázových transakcí je aplikace velmi rychlá a příliš nezatěžuje server. Ukládání nových dat z výstupu NfDumpu tedy probíhá automaticky v nastavený interval.

Vlastní portál pro zobrazení předpřipravených statistik je implementován v jazyce PHP, XHTML s použitím kaskádových stylů CSS a JavaScriptu, v kterém jsou implementovány interaktivní grafy. Výsledným řešením je jednoduché rozhraní umožňující měnit intervaly a rozsahy navržených typů statistik. Obsahuje také jednoduchou správu uživatelských účtů. Celý portál je chráněn heslem a je třeba mít vytvořený účet pro vstup do portálu, čímž je splněn i požadavek dostupnosti z internetu a bezpečnosti proti nezvaným hostům.

Všechny body zadání, jakožto i nároky, které jsem na tuto práci sám kladl, se mi podařilo splnit. Při práci, která byla zaměřena širším směrem, jsem využil znalosti hned z několika předmětů, které jsem v předchozích letech studoval, a zhodnotil dosavadní i nově nabité zkušenosti v praxi.

Z hlediska budoucího vývoje je portál postaven na objektovém návrhu a je počítáno s implementací dalších vylepšení v oblasti vizualizace a uživatelského rozhraní. Snadno je také možné přidat další typy statistik.

Literatura

[1] Wikipedie, otevřená encyklopedie: NetFlow [online]. 26.3.2009 [cit. 25.4.2009]. Dostupné na URL: <http://cs.wikipedia.org/wiki/Netflow>

[2] ŽÁDNÍK, Martin. Síťové aplikace a správa sítí 2007: NetFlow [online]. 2007 [cit. 26.4.2009] Dostupné na URL: <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ISA-IT/lectures/archiv-2007/isa_netflow.pdf>

[3] SourceForge: NFDUMP [online]. 15.8.2007 [cit. 26.4.2009] Dostupné na URL: <http://nfdump.sourceforge.net/>

[4] Sbírka zákonů: Předpis č. 485/2005 [online]. 15.12.2005 [cit. 26.4.2009] Dostupné na URL: <http://www.sagit.cz/pages/sbirkatxt.asp?zdroj=sb05485&cd=76&typ=r>

[5] World Wide Web Consorcitum: XHTML, CSS [online]. 29.1.2009 [cit. 26.4.2009] Dostupné na URL: <http://www.w3.org/>

[6] HRUŠKA, Tomáš; BURGET, Radek: *Internetové aplikace IV. - část programování serveru (PHP)* © 2006-2007. 1.2.2007 [cit. 26.4.2009] Dostupné na URL: <https://www.fit.vutbr.cz/study/courses/WAP/private/opory OporaWAP4ProgramovaniServeru.pdf>

[7] HRUŠKA, Tomáš: *Internetové aplikace VI. - část programování klienta (JavaScript)* © 2006-2007. 1.2.2007 [cit. 26.4.2009] Dostupné na URL: <https://www.fit.vutbr.cz/study/courses/WAP/private/opory/ OporaWAP6ProgramovaniKlienta.pdf>

[8] LUPA: Útoky typu DoS [online]. 19.9.2006 [cit.30.4.2009] Dostupné na URL: <http://www.lupa.cz/serialy/utoky-typu-dos/>

[9] PUŽMANOVÁ, Rita: *Moderní komunikační sítě od A do Z*, nakladatelství Computer Press, II. vydání, 2006, s. 432, ISBN: 80-251-1278-0.

[10] Google Visualization API: Documentation [online]. [cit. 27.4.2009] Dostupé na URL: <http://code.google.com/intl/cs/apis/visualization/>

[11] SQLite Home Page: About SQLite [online]. 28.3.2009 [cit. 26.4.2009] Dostupné na URL: <http://www.sqlite.org/>

[12] Jak psát web: HMTL, CSS [online] Dostupné na URL: <http://www.jakpsatweb.cz/>

[13] PHP Manuál: [online]. Dostupné na URL: <http://www.php.net/manual/cs/>

[14] Interval webdesign: HTML, OOP v PHP, Cron, JavaScript [online]. Dostupné na URL: <http://www.interval.cz/>

[15] FlowMon - kompletní řřešení pro monitorování sítí na bázi NetFlow - INVEA-TECH [online].
Dostupné na URL: <http://www.invea.cz/cs/products/flowmon>

[16] NetFlow Monitor: [online].
Dostupné na URL: <http://netflow.cesnet.cz/>

[17] CAIDA tools: cflowd, FlowScan [online].
Dostupné na URL: <http://www.caida.org/tools/measurement/cflowd/>

[18] inMon Corporation: , sFlow, sFlowTrend [online].
Dostupné na URL: <http://www.inmon.com/products/sFlowTrend.php>

[19] Free NetFLow tools: fprobe, flowscan, Scrutinizer, Stager [online]
Dostupné na URL: <http://www.networkuptime.com/tools/netflow/>

[20] Firewall.cz – TCP analysis [online].
Dostupné na URL: <http://www.firewall.cx/tcp-analysis-section-4.php>

[21] Owebu: Skenování portů [online]. 4.10.2004 [cit. 14.5.2009]
Dostupné na URL: <http://www.owebu.cz/bezpecnost/vypis.php?clanek=365>

Seznam příloh

Příloha 1. DVD obsahující zdrojové kódy, testovací data, soubor README.TXT a programovou dokumentaci.