VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

OCHRANA DATOVÉ SÍTĚ S VYUŽITÍM NETFLOW DAT

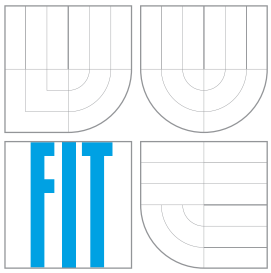
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB ČEGAN

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

OCHRANA DATOVÉ SÍTĚ S VYUŽITÍM NETFLOW DAT

NETWORK PROTECTION USING NETFLOW DATA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAKUB ČEGAN
AUTHOR

VEDOUCÍ PRÁCE — Ing. JIŘÍ TOBOLA
SUPERVISOR

BRNO 2009

Abstrakt

Tato práce popisuje zabezpečení datové sítě pomocí technologie NetFlow. V úvodu jsou popsány některé možné hrozby, které mohou datovou síť postihnout a které jsou rozpoznatelné pomocí NetFlow dat. V další části práce jsou navržena určitá detekční pravidla a dvoukrokový způsob detekce pomocí jejich použití. Tato pravidla byla formulována na základě pozorování a experimentů v datové síti.

Abstract

This thesis deals with the using of NetFlow data for computer network protection. First are described some types of network security threats. After study of these threats and many experiments were designed detection rules for them. New detection form were designed too. It is working with two step detection of threats.

Klíčová slova

NetFlow, detekce, ochrana datové sítě, BitTorrent, skenování, DDoS útok

Keywords

NetFlow, detection, Network Protection, BitTorrent, Scans, DDoS attack

Citace

Jakub Čegan: Ochrana datové sítě s využitím NetFlow dat, bakalářská práce, Brno, FIT VUT v Brně, 2009

Ochrana datové sítě s využitím NetFlow dat

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Toboly. Další informace mi poskytl RNDr. Pavel Minařík.

.....
Jakub Čegan
18. května 2009

Poděkování

Rád bych poděkoval vedoucímu své práce Ing. Jiřímu Tobolovi za čas věnovaný konzultacím této práce a za to, že mi umožnil přístup k počítačové síti, kde prováděl své testy. Také bych rád poděkoval RNDr. Pavlu Minaříkovi za jeho čas strávený se mnou při pro mne důležitých konzultacích s podnětnou diskuzí. V neposlední řadě mé poděkování patří Ing. Janu Pazderovi za pomoc při správě testovacího stroje.

Obsah

Kapitola 1

Úvod

V dnešní době dochází k neustálému rozvoji datových sítí. Dramaticky se zvyšuje jejich počet a velikost. Dochází také k masivnímu nárůstu rychlosti sítí. S rozšiřováním datových sítí roste potřeba jejich ochrany před snahami o narušení jak z vnějšího světa, tak od uživatelů uvnitř sledované sítě. Je třeba účinnou formou prosazovat zákony vztahující se na chování v datové síti a také na data, která se v ní nacházejí. Je tedy nezbytné neustále kontrolovat dění v síti, a sledovat jestli nedochází k protiprávnímu jednání. Na druhou stranu je ovšem nutné dodržovat právo na soukromí uživatelů.

Vzhledem ke zrychlování sítí a zvětšování objemu dat jimi protékajících je zpracovávání každého paketu velmi náročné a v konečném důsledku mohou běžná monitorovací zařízení síť zbytečně zatěžovat a stát se tak jejím úzkým hrdlem. Zabezpečení takové datové sítě pomocí NetFlow umožňuje v případě použití pasivních sond sledovat provoz v síti bez toho, aniž by byla síť zbytečně zatěžována.

Tato technologie může být využita poskytovateli internetu pro zajištění účtovatelnosti poskytovaných dat a dohledu nad dodržováním podmínek stanovaných smlouvou. Stejně tak mohou být NetFlow data použita v IT odděleních velkých firem k monitorování komunikace v jejich vnitřní síti. Pomocí statistické analýzy je možné vytvořit profily jednotlivých uživatelů a sledovat tak jejich chování bez narušení jejich soukromí. Dále je možné zabezpečit síť před úniky dat, napadením DoS a DDoS útoky a před šířením ilegálních dat.

Předmětem této práce je vypracování náhledu na bezpečnost datové sítě skrze použití technologie NetFlow. V první části je stručně nastíněn pohled na dnes běžně používaná zabezpečení datové sítě. V následující kapitole je rozebráno monitorování počítačové sítě pomocí technologie NetFlow. Je zde nastíněn stručný úvod do NetFlow protokolu a také je zde uveden přehled programů a jejich funkcí pro práci s daty se zaměřením na program použitý v této práci. Poté následuje rozbor několika hrozeb pro datovou síť s ohledem na jejich projevy v NetFlow datech s uvedenými možnostmi jejich detekce. Poslední částí práce je rozbor dosažených výsledků a nastínění možnost pokračování v práci.

Kapitola 2

Bezpečnost datových sítí

2.1 Firewall

Firewall je aplikace, či síťové zařízení umožňující zabezpečení před hrozbami přicházejícími do důvěryhodné vnitřní sítě z nedůvěryhodné sítě vnější. Jedná se o bod na přístupové lince nebo linkách, přes který prochází veškerá komunikace mezi sítěmi. V tomto místě dochází ke kontrole dle zadané bezpečnostní politiky a k propouštění, případně zamítání spojení. Použité technologie firewallu je možné rozdělit podle doby vzniku a přístupu k problému do tří níže popsaných kategorií [17].

2.1.1 Paketové filtry

Paketové filtry jsou prvním a nejstarším druhem firewallu. Fungují na principu aplikace pravidel pro povolení nebo zakázání každého spojení procházejícího firewallem. Tato síťová spojení jsou identifikována dle zdrojové a cílové ip adresy, čísel portů a protokolu. Výhodou tohoto typu firewallu je rychlost práce s daty.

Zásadní nevýhodou pak je bezstavovost, tedy nemožnost rozhodnout, zda paket patří k nějakému již existujícímu spojení [1].

2.1.2 Stavové filtry

Stavové filtry fungují obdobně jako paketové filtry, ovšem s přidanou funkcionalitou ukládání povolených spojeních. Tato schopnost umožňuje firewallu rozhodnout, zda je příchozí paket součástí již existujícího spojení, nebo se jedná o nové spojení a je tedy třeba o něm teprve rozhodnout.

Výhodou stavových filtrů je vyšší rychlost a bezpečnost než u předchozích paketových firewallů.

2.1.3 Firewally aplikační vrstvy

Tyto firewally také nazývané proxy firewally vznikly jako poslední. Jak již název napovídá, pracují na sedmé vrstvě ISO/OSI modelu [1]. Dochází zde k úplnému oddělení spojení, protože se zde oba počítače přímo nepropojují. Nejdříve je navázáno spojení z počítače inicializujícího komunikaci s proxy firewallem. Poté, pokud je spojení povoleno, dojde k propojení z firewallu na cílový počítač a data jsou předána v původní podobě. Vzhledem k principu spojení tyto firewally také fungují jako překladače síťových adres (NAT).

Výhodou tohoto přístupu je vysoký stupeň ochrany sítě. Nevýhodou jsou vysoké nároky na hardware firewallu a relativní pomalost oproti výše uvedeným přístupům.

2.2 Intrusion detection system (IDS)

Jedná se o softwarový či hardwarový systém pro detekci nežádoucího chování v datové síti. Systém dokáže sledovat obsah paketů a tak odhalit pokus o narušení bezpečnosti přicházející jak z vnějšího prostředí, typicky internetu, tak i zevnitř sítě. Jeho hlavními úkoly jsou především detekce útoků proti zranitelným službám a aplikacím, detekce pokusu o získání přístupových práv k systémům a v neposlední řadě detekce malwaru, tj. trojských koní, virů a červů. Systémy pro detekci vniknutí je možné rozdělit do následujících dvou základních kategorií [14]:

Network intrusion detection system (NIDS) Senzory NIDS bývají umístěny u vstupních bodů do počítačové sítě, případně u vstupu do demilitarizované zóny této sítě. Dochází zde k zachycování veškerého síťového provozu a vyhledávání nežádoucích projevů. Detekce je založena na hledání vzorků v paketech uvedených v databázi systému jako příznačné pro útok, nebo odhalování neobvyklé aktivity paketů, která je příznakem probíhajícího útoku [14].

Host intrusion detection system (HIDS) HIDS se od výše uvedených NIDS liší tím, že jsou vždy instalovány na určitý počítač a zajišťují nad ním nepřetržitý dohled. Nedochází zde ke sledování paketů, ale dle projevů chování jednotlivých uživatelů systém zjišťuje, jestli nedochází k narušení bezpečnosti. Může se jednat o pokusy o přihlášení do systému neautorizovanými uživately, či krajně podezřelé chování oprávněných uživatelů.

Nevýhodou IDS je, že při rychlostech linky kolem 1Gb/s a výše, již přestávají tyto systémy zvládat zpracování obrovského množství procházejících paketů a stávají se tak úzkým hrdlem. Použitelnost IDS na vysoko rychlostních linkách je možné zajistit pomocí jejich hardwarové akcelerace [26]. Vychází se zde z předpokladu, že většina paketů patří legitimnímu provozu a jsou tudíž neškodné. Pakety jsou na úrovni hardwaru porovnávány s databází nezávadných paketů. V případě shody jsou zahozeny a do IDS se tedy dostanou pouze pakety potencionálně patřící k útoku.

2.3 Intrusion prevention system (IPS)

V tomto případě se jedná o rozšíření již dříve zmíněného systému IDS. IPS dokáže nejen útoky detekovat, ale také narozdíl od jeho předchůdce, se jim také aktivně bránit. Tuto obranu proti útokům zajišťují dvě různé techniky. První je zrušení probíhající komunikace pomocí odeslání TCP paketu s nastaveným příznakem RST, případně zaslání ICMP zprávy Unreachable útočníkovi [14] a tím ukončení probíhajícího spojení. Tato metoda má anglický název sniping. Druhou technikou je nařízení vstupnímu firewallu nebo směrovači, aby začal odmítat detekovanou závadnou komunikaci (shunning) [14]. IPS se dělí na Host intrusion prevention system (HIPS) a Network intrusion prevention system (NIPS) dle podobné logiky jako IDS.

2.4 Honeypot

Honeypot, neboli návnada je součástí zabezpečení datové sítě, která má za úkol odlákat útočníka od skutečných systémů a svést jej na falešnou stopu. Většinou bývá umístěna v demilitarizované zóně sítě. Dalším úkolem návnady bývá sběr informací o útočnících a jejich technikách. Tyto informace jsou poté využity pro vylepšování technik obrany před útoky a v případě předání incidentu orgánům činným v trestním řízení mohou posloužit jako důkazní materiál. Návnady se dělí dle své funkce a složitosti provedení na tři typy [14]:

Monitory portů Jedná se o nejjednodušší typ návnady. Poslouchají na portech často vyhledávaných útočníky a umožňují jejich připojení. Pokusy o připojení poté zaznamenávají.

Falešné systémy Falešné systémy jdou o krok dále než Monitory portů. Předstírají, že se jedná o plnohodnotný systém se všemi nezbytně nutnými projevy takového systému a zvyšují tak šanci, že útočník uvěří, že se jedná o skutečný systém a zahájí útok.

Násobné falešné systémy Jedná se o další rozvedení myšlenky návnad. Násobné falešné systémy umějí kromě předstírání několika různých služeb také předstírat různé operační systémy.

2.5 SNMP protokol

Protokol SNMP, celým názvem Simple Network Management Protocol, je standardizovaným protokolem sloužícím pro správu počítačové sítě. K transportu dat používá protokol UDP. Od verze 3 SNMP podporuje také autentizaci a šifrování. Umožňuje sběr důležitých dat a jejich následné vyhodnocení. V tomto protokolu vystupují dvě entity a to Agent a Manager. Agent příjímá požadavky od Managera a posílá mu zpět posbíraná dat. Manager příjímá data od a Agentů, která poté ukládá a následně zpracovává.

2.6 Antispamová ochrana

Existuje mnoho druhů spamu, ale pro účely této práce je termínem Spam míněn spam šířený pomocí emailů. Spam je jedním ze zásadních problémů dnešního internetu. Některé zdroje uvádějí, že až 79% veškeré emailové komunikace je spam [4]. Naprostou většinu z něj mají na svědomí počítače nedobrovolně zapojené ve velkých sítích, zvaných botnety [9]. Ochrana proti spamu je možná na více úrovních. Pro tuto práci je důležitý způsob zabraňující přijetí již odeslaných spamů.

2.6.1 Filtrace dle odesilatele

Blacklisting Blacklisting je jednoduchý způsob filtrování spamu, založený na důvěryhodnosti adresy odesilatele nebo ještě lépe na důvěryhodnosti ip adresy poštovního serveru. Pokud bylo zaznamenáno šíření spamu z určitého serveru, pak je jeho adresa přidána do Blacklistu a emaily přícházející z ní jsou buď přímo odmítnuty nebo označeny jako spam.

Greylisting Jedná se vylepšení Blacklistingu pomocí dynamického chování. První příchozí zpráva z emailového serveru je pozdržena a serveru je vrácena informace o nemožnosti

ji dočasně doručit. Pokud dojde k opětovnému pokusu doručit zprávu, filtr ji propustí a server je na určitou dobu označen jako důvěryhodný. Další zprávy již procházejí po dobu trvání důvěry bez zdržení. Po uplynutí nastavené doby, po kterou je server označen jako důvěryhodný, se proces opakuje. Je zde využito faktu, že spam roboti se emaily většinou nepokoušejí znovu doručit.

2.6.2 Filtrace dle obsahu zprávy

Filtrace pomocí pravidel Tento postup je založen na filtraci spamu dle pro něj charakteristických vlastností. Mezi tyto vlastnosti patří například typická klíčová slova nebo slovní spojení. V závislosti na počtu splněných kritérií je poté zpráva případně označena jako spam.

Bayesovské filtry Tyto filtry jsou založeny na učení a umělé inteligenci [10]. Filtrům jsou předkládány emaily označené jako spam a *nespam* (ham), pomocí kterých se učí obsah nevyžádaných zpráv. Při vlastní práci je filtru předložen obsah emailu a následně je dle naučených pravidel rozhodnuto o jeho osudu.

Kapitola 3

Monitorování provozu na základě síťových toků

3.1 Úvod do NetFlow

NetFlow je otevřený protokol pro přenos informací o tocích v datové síti původně vyvinutý společností Cisco Systems jako doplňková služba pro jejich směrovače. Definici toku (flow) v NetFlow lze vyjádřit tak, že se jedná o jednosměrnou posloupnost paketů se shodnou zdrojovou a cílovou ip adresou, zdrojovým a cílovým portem, protokolem (TCP, UDP, ICMP, IGMP), ToS a číslem rozhraní [11]. V NetFlow datech jsou zaneseny veškeré informace o spojení, doba jeho vzniku, délka trvání, počet přenesených paketů a bytů a další údaje. Není zde však uložen vlastní obsah paketů.

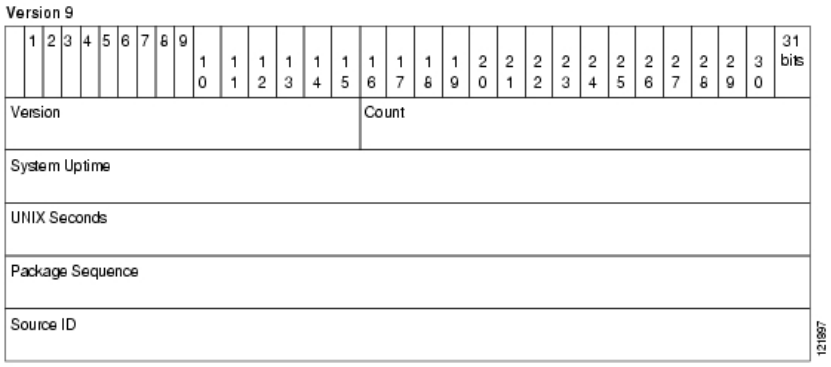
Tradiční architektura dle společnosti Cisco dříve vypadala tak, že směrovače v síti kromě své běžné funkce také sbíraly NetFlow data a počítaly z nich statistiky. Taková architektura ovšem při větším provozu způsobovala velké vytížení směrovačů a to až do té míry, že mohlo dojít k omezení jejich primární funkce. Proto se přistoupilo k použití specializovaného hardwaru, tzv. pasivních sond. Tyto sondy je možné zapojit do libovolného místa v sítí. Nejčastěji se jedná o přístupové linky do sítě, případně její klíčové uzly. Sondy data přes ně procházející pouze monitorují a nijak do nich nezasahují, proto se nazývají pasivní. Statistiky jsou poté odesílány oddělenou linkou do kolektoru a ve sledovaném provozu se tedy vůbec neprojeví. Z tohoto důvodu je velmi složité sondy odhalit a jsou tedy obtížným cílem pro útočníky. Dvě podstatné části NetFlow architektury jsou nazývány Exportér a Kolektor.

Exportér Jak již bylo výše uvedeno, jedná se buď o pasivní sondu, router nebo softwarovou implementaci [11]. Příchozí paket je Exportérem přijat a jsou z něj extrahovány požadované informace. Poté je vyhledáno, zda paket patří do již existujícího flow, které je aktualizováno, nebo je založeno flow nové [11].

Kolektor Kolektor je zařízení s velkou úložnou kapacitou, jež sbírá data z několika exportérů. Nad datovým úložištěm většinou běží aplikace umožňující operace s NetFlow daty. Mezi běžné operace patří filtrování toků na základě pravidel, agregace toků dle zadaných kritérií a zobrazování dat.

První masově použitou verzí NetFlow protokolu byla verze 5. Protokol verze 9 má na rozdíl od předchozích verzí svou strukturu danou šablonou. Hlavní výhodou tohoto přístupu je, že umožňuje rozšíření NetFlow služeb bez změny původního záznamu [5]. Dále je umožněn

záznam ip adres ve formátu IPv6 a je tak pamatováno na připravovanou změnu po vyčerpání adres IPv4. Obrázek 3.1 zachycuje hlavičku NetFlow paketu v9. Na základě protokolu verze 9 vznikl nový prototokol Internet Protocol Information Export (IPFIX). Jedná se de facto o NetFlow verze 10, které bylo prohlášeno za nový IETF standard [12], aby došlo ke sjednocení různých metod používaných k práci s ip flow.



Obrázek 3.1: Hlavička NetFlow paketu v9 [19]

Hlavní výhodou, která hovoří pro použití NetFlow, je fakt, že tato technologie je schopna fungovat i na sítích o rychlostech 10 Gb/s a na sítích s velkým provozem, kde by bylo nasazení jiných technologií nemožné nebo velmi obtížné. Pomocí informací získaných z NetFlow dat je možné tvořit téměř libovolné statistiky. Na základě těchto statistik je možné tvořit profily chování jednotlivých strojů zapojených do sítě a při zjištěné odchylce spustit poplach. Silnou stránkou NetFlow je také jeho imunita proti šifrovanému provozu, která vychází ze skutečnosti, že není pracováno s obsahem paketů, ale s chrakteristikou jednotlivých datových toků.

3.2 Software pro zpracování NetFlow dat

Na vývoj aplikací pro práci s NetFlow daty se dnes soustředí mnoho společností. Jedná se buď o komplexní komerční řešení od renomovaných firem, jako je Hewlett Packard nebo IBM a nebo je možné nalézt na internetu několik velmi dobrých kompletních open source řešení. Dále lze nalézt mnoho programů pro analýzu NetFlow dat a grafických rozhraní pro kolektory. Stručný přehled dostupných softwarů lze shlédnout v přiložené tabulce 3.1. Podrobnější informace je možné najít na následujícím odkazu [20].

Mezi nejzajímavější z výše uvedených aplikací patří díky svým vlastnostem následující programy:

Peak Flow Tento komerční produkt společnosti Arbor Networks umožňuje běžný sběr a analýzu NetFlow dat. V oblasti bezpečnosti je zaměřen na zjišťování p2p sítí, monitorování messengerů a odhalování DDoS útoků.

StealthWatch Řešení od společnosti Lancope, technologického partnera Cisco Systems, využívá pro svou detekci bezpečnostních rizik statistickou analýzu NetFlow dat.

NfSen, NfDump Tyto programy byly jako jedno z nejlepších dostupných open source řešení použity pro vypracováni této práce a jsou podrobněji rozebrány dále.

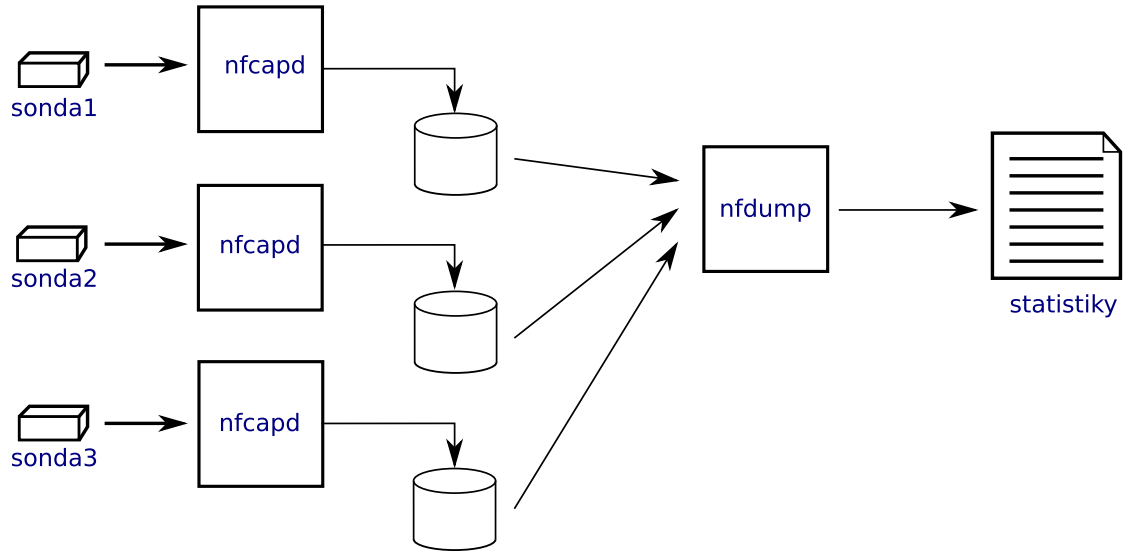
Produkt	Výrobce	Licence
NetFlow Monitor	Cesnet	freeware
StealthWatch	Lancope	komerční
NfSen, NfDump	Peter Haag, SWITCH	freeware
Flow Tools	Splintered	freeware
PeakFlow	Arbor Networks	komerční
Flow Viewer	NASA	freeware
NetFlow Insight	Hewlett Packard	komerční
Aurora	IBM	komerční
NetFlow Analyzer	AdventNet	komerční
NTOP	Ethereal	freeware

Tabulka 3.1: Stručný přehled softwaru pro práci s NetFlow daty

Flow Tools Jedná se o balíček programů velmi podobný balíku NfDump. Jeho největší nevýhodou je, že není schopen pracovat s NetFlow v9, ale zvládá pouze starší verzi 8. Tento softwarový balík obsahuje software pro zachytávaní NetFlow dat a několik programů pro jejich úpravu. Mezi zajímavé součásti balíku patří utilita flow-dscan, která umožňuje jednoduše v NetFlow datech vyhledávat některé typy skenů a DoS útoků.

3.2.1 NfDump a NfSen

Nfdump je balíček nástrojů určený pro sběr a zpracování NetFlow verze 5, verze 7 a verze 9. Software je distribuován pod BSD licencí a je spustitelný na všech BSD a Posix platformách [21]. Cílem tohoto balíčku je, aby bylo možné stejně jednoduše prohledávat již proběhlá flow, stejně jako zajímavá aktuální flow. Princip funkce jednotlivých programů z balíčku při sběru NetFlow dat je zobrazen na obrázku 3.2.



Obrázek 3.2: Zpracování dat pomocí NfDump

nfcapd Démon nfcapd čte data a ukládá je do souborů s určitou časovou periodou. Nejčastěji se jedná o pětiminutovou periodu. Nfcapd čte NetFlow data verze 5,7 a 9. Pro každý NetFlow stream je třeba jeden běžící nfcapd.

nfdump Program nfdump čte data uložená programem nfcapd. Pomocí tohoto programu je možné filtrovat data podle různých pravidel a generovat statistiky nejvýraznějších flow. Je možné vybrat si z několika úrovní detailu výpisů nebo je možné si nastavit vlastní formát vypisovaných dat. Toto je velmi užitečné pro použití ve skriptech. Více o zápise pravidel je možné zjistit na stránce projektu nfdump [21].

Grafický frontend programu NfDump nese jméno NfSen. Jedná se o webové rozhraní napsané v jazyce PHP a je opět šířeno pod BSD licencí. Umožňuje zobrazovat grafy NetFlow dat za pomoci Round Robin Databáze. Veškeré grafy v této práci zobrazující datové toky pocházejí právě z programu NfSen. Dále je možné procházet NetFlow data v určitém časovém rozmezí, nastavovat upozornění na události a také doplňovat do aplikace vlastní pluginy [22].

3.3 Statistické metody detekce anomálií

Běžně používané statistických způsoby detekce jsou založeny na vytvoření profilu chování jednotlivých počítačů v datové síti. Používají se zde různě velké způsoby agregace NetFlow dat. Na tato data jsou poté aplikovány statistické metody. Pomocí statistických metod je možné zjišťovat odchylky provozu od normálního stavu. Pokud profilu počítače odpovídá několik připojení na emailový server za den, pak může nárůst na několik stovek spojení za den znamenat, že počítač byl nakažen některým z červů a nyní je součástí botnetu šířícího spam. Bohužel ne všechny hrozby v datové síti jsou těmito metodami odhalitelné.

Metoda Minds Minnesota Intrusion Detection System porovnává v rámci časových úseků toky s předcházejícími průměrnými hodnotami. Tyto toky jsou agregované dle zdrojové a cílové ip adresy a zdrojového a cílového portu.

Metoda Xu et al. Tato metoda je pojmenována dle svých autorů. Metoda určuje entropii cílové ip adresy, cílového portu a zdrojového portu jako sadu všech spojení vycházejících z každé zdrojové ip adresy [13]. Poté jsou jednotlivá spojení klasifikována do tříd dle míry své entropie [13].

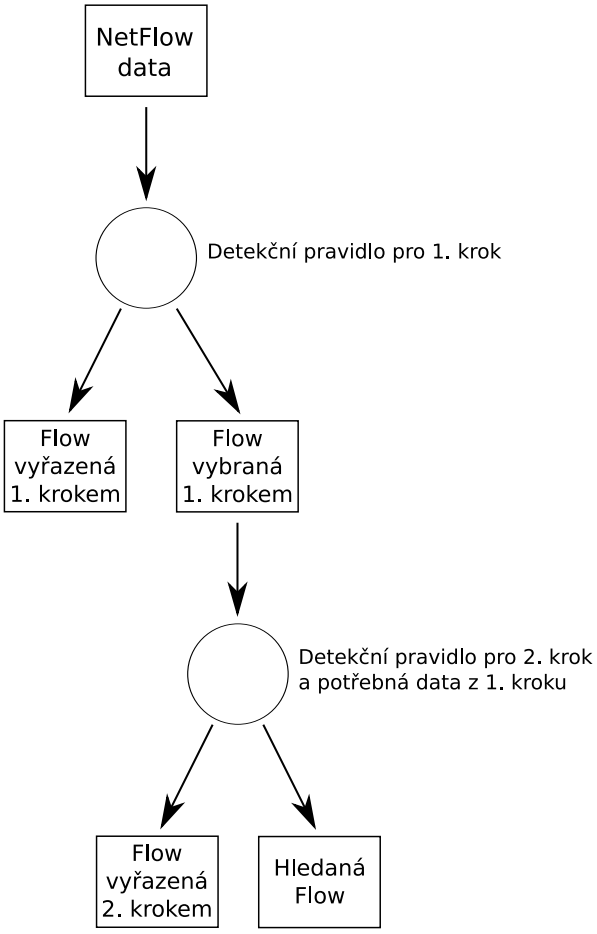
Metoda Volume Prediction Metoda Volume Prediction postupně počítá počty toků, bytů a paketů pro jednotlivé ip adresy v datové síti a kontroluje je oproti hodnotám v určitém časovém úseku [13]. Odychylka naměřených hodnot od hodnot předpovězených pomocí dat z minulosti, představuje míru anomálie [13]. Model se v dalším kroku průběžně zaktualizuje pomocí nově naměřených hodnot, čímž se adaptuje na změny chování sítě.

3.4 Detekce pomocí pravidel

Narozdíl od výše zmíněného způsobu detekce anomálií je detekce navržená v této práci založena na aplikaci jednoduchých pravidel. Tato pravidla jsou vytvořena na míru každé hrozbě na základě pozorování jejích projevů v NetFlow datech v testovací síti a nalezení pro ni charakteristického datového toku nebo toků.

V případech, kdy je detekce jevu triviální, nebo je potřeba detekovat jev jen obecně, postačuje použití jednho detekčního pravidla. Pokud se jedná o komplikovanější problém, nebo je třeba dosáhnout lepší přesnosti, pak je potřeba použít dvě nebo obecně více detekčních pravidel. Tímto se zvyšuje účinnost detekce a omezuje se počet falešných poplachů. Komunikace musí totiž odpovídat detekčním pravidlům na více místech a je tak omezena možnost náhodné shody.

Detekce se provede tak, že program nfdump je nejprve spuštěn s prvním pravidlem. Čím přesněji je toto pravidlo nastaveno, tím rychleji proběhne druhý krok detekce, protože bude zpracovávat méně dat. Výstupem je soubor určitých flow. Z těchto flow jsou vyňaty údaje identifikující spojení. Jedná se o zdrojovou a cílovou ip adresu a zdrojový a cílový port. V případě, že druhé pravidlo slouží k vyhledání flow, které je odpovědí na flow vyhledané v prvním kroku, dojde k výměně údajů o zdroji a cíli. Ze zdrojového portu a ip adresy se stane cílový port a ip adresa a naopak. Tato data jsou poté přidána k druhému detekčnímu pravidlu a program nfdump je spuštěn znovu pro každé dříve nalezené spojení. Jako platná jsou brána pouze flow s počátečním časem větším než čas původního flow, ze kterého byly získány informace o spojení. V případě, že jsou nalezena platná flow, jedná se o potvrzení hledaného jevu. V opačném případě jde o falešný poplach. Princip detekce je zachycen na obrázku 3.3.



Obrázek 3.3: Diagram detekce pomocí dvou pravidel

Kapitola 4

Hrozby pro datovou síť

4.1 Skenování portů

Skenování portů je velmi častým jevem, zvláště u počítačů přímo připojených k internetu. Je obtížné rozhodnout o závažnosti tohoto činu. Na jedné straně dochází ke skenování počítačů vlastně neustále bez vážnějších následků. Na druhé straně si ovšem útočník může připravovat prostor pro větší útok nebo se může poohlížet po špatně zabezpečeném systému, nebo nechráněném portu, který používá pro něj zajímavá zranitelná služba.

4.1.1 TCP SYN sken

SYN sken je velmi oblíbený sken. Za jeho oblibu může hlavně jeho jednoduchost a rychlost. Tento sken zneužívá mechanizmus, jakým jsou uzavírána spojení TCP protokolu, který se nazývá Třícestný handshake (Three-Way Handshake). Na cílový port je odeslán paket s nastaveným příznakem synchronizace (SYN paket). Pokud je skenovaný port uzavřen, je zpět odeslán paket s příznakem reset (RST). V případě otevřeného portu je zpět na skenující počítač odesláná odpověď SYN/ACK a skenovaný počítač čeká na potvrzení paketem s nastaveným příznakem ACK od útočníka, aby mohlo být ustaveno spojení. Paket ovšem nikdy nepřijde a spojení je po chvíli zahozeno, nebo je zpět místo příznaku ACK odeslán paket s příznakem RST, který spojení ukončí [8].

SYN sken se v NetFlow datech projevuje jako velký počet toků se stejnými cílovými ip adresami a rozdílnými čísly portů, které proběhly v krátkém časovém okamžiku. Zdrojové ip adresy jsou ve většině případů u všech toků stejné, protože se útočník nesnaží zamaskovat své konání pomocí falešných skenů. Při použití falešných skenů jsou zdrojové adresy rozdílné. V obou případech mají toky stejnou velikost a počet paketů přicházejících na testovaný port se rovná jedné. Pro odhalení skenu je možné použít filtr:

```
proto TCP and (flags S and not flags ARPFU)
```

4.1.2 FIN sken, Xmas sken a Null sken

Tyto skeny mají na rozdíl od SYN skenu tu výhodu, že mohou proklouznout skrz nestavové firewally a jednoduché paketové filtry na routerech, což činí tyto skeny o něco méně nápadné. Velkou nevýhodou je, že ne každý systém je postaven dle normy RFC 793 [3], na kterou se tyto skeny spoléhají, a tak výsledky závisí na faktu, jestli testovaný systém je v souladu s normou. Systémy postavené přesně dle normy odesílají paket s příznakem

RST pro uzavřené porty. Pokud není přijata odpověď, je port označen jako otevřený nebo filtrovaný.

Také pravidla pro detekci těchto skenů se liší pouze v malém detailu a to, jak je uvedeno níže, v nastavených příznacích.

FIN sken FIN sken se používá ke zjištění stavu portů na počítačích s operačním systémem Unix. Jak již název napovídá, je odeslán paket s nastaveným příznakem FIN. Pravidlo pro detekci je zapsáno ve tvaru:

```
proto TCP and (flags F and not flags PARUS)
```

Xmas sken Při Xmas skenu je na testovaný port odeslán paket s nastavenými flagy FIN, PSH a URG. Pravidlo pro detekci je zapsáno ve tvaru:

```
proto TCP and (flags FUP and not flags ARS)
```

Null sken Na cílový počítač je odeslán paket s vynulovanými příznaky. Pravidlo pro detekci je zapsáno ve tvaru:

```
proto TCP and not flags FUPARS
```

4.2 Detekce malware

Malware je souhrné označení pro veškerý škodlivý software, jmenovitě se jedná o viry, červy a trojské koně. Tyto škodlivé kódy jsou šířeny za různým účelem. Může jít o obyčejný vandalismus nebo v dnešní době častěji o útok zločineckých organizací se záměrem získat od uživatelů citlivá data, jako jsou například hesla k internetovému bankovnictví a dalším službám. Také může jít o pokus zapojit napadený počítač do některého z mnoha botnetů zodpovědných za šíření spamu a DDoS útoky.

Boj proti malwaru je za pomoci NetFlow dat poněkud obtížný. Protože se malware šíří hlavně pomocí emailů a závadných internetových stránek, není možné s výjimkou použití blacklistů zabránit stažení tohoto škodlivého kódu a použití NetFlow dat tedy rozhodně v žádném případě nenahrazuje antivirové a antimalware programy. Je ovšem možné pomocí jak statistických metod detekce, tak detekčních pravidel vysledovat malwarem infikované počítače. NetFlow data tedy umožní rychlou reakci a pomohou minimalizovat škody způsobené infikovaným počítačem.

K nejčastějším projevům malware v NetFlow datech patří například extrémní zvýšení spojení s SMTP servery. Dále nárůst počtu skenů z počítačů ve vnitřní síti, kdy se červi pokoušejí nalézt další zranitelné stroje, aby je mohli infikovat. Také je možné detekovat zombie počítače zapojené do botnetů pomocí odhalení jejich komunikace s botmasterem. Tato komunikace bývá nejčastěji založena na protokolu IRC, či HTTP. Výjimkou ovšem nejsou ani botnety schopné komunikovat na libovolném portu a svou komunikaci šifrovat. Botnety v NetFlow datech je možné odhalovat jak pomocí statistických metod, tak pomocí nalezení vhodného vzorku komunikace ve flow a vytvoření detekčního pravidla.

4.3 Útok na SSH

SSH je program a také protokol určený pro zabezpečený vzdálený přístup k systému. Umožňuje autentizaci a šifrování. Byl vyvinut za účelem nahrazení programů komunikujících nezabezpečenou cestou, jako je například telnet.

Protože SSH umožňuje přistoupit do systému vzdáleně, bývá častým cílem útočníků, kteří se pokouší získat přístupové heslo, tím získat přístup do systému a ten následně ovládnout nebo poškodit. Pro znesnadnění detekce mohou být tyto útoky vedeny z několika míst, případně mohou být v čase rozloženy tak, aby budily co nejmenší podezření. Proto by v ideálním případě měly být účty s vysokými právy jako je například root přes SSH nedostupné. Útoky proti heslům SSH účtů lze rozdělit stejně jako veškeré útoky proti heslům na dva druhy.

Útok hrubou silou je v podstatě téměř nepoužitelný pro svoji extrémní časovou náročnost. Dochází zde ke zkoušení všech kombinací znaků ze zadané množiny, což u hesla, které má 5 znaků a je složeno pouze z malých písmen, představuje zhruba 12,4 milionu kombinací.

Druhým používaným útokem je útok slovníkový. Heslo k účtu je testováno proti rozsáhlé databázi slov, takzavnému slovníku. Úspěšnost útoku silně závisí na dokonalosti a obsáhlosti slovníku a také samozřejmě na tom, zda je k tomuto typu útoku heslo náchylné. Mělo by se jednat o smysluplné slovo bez speciálních znaků (/*-+,.?).

Útok na SSH se v NetFlow datech projevuje jako velký počet toků vykazujících velkou míru podobnosti v krátkém časovém úseku. Při jeho hledání je možné se omezit na port 22, kde bývá defaultně spuštěn SSH server. Jednotlivé toky směrem od útočníka mají vždy více než 10 paketů, protože při nižším počtu nemůže jít o korektní přihlášení [16]. Ve většině pozorovaných případů při tvorbě této práce měla flow jdoucí směrem od útočníka velikost mezi 1100-1400 byty a počet paketů 12-15. Odpovědi serveru mezi sebou vykazovaly větší podobnost než útočníkovy požadavky. Počet paketů se zde pohyboval mezi 13-16 a velikost byla 2300-3200 bytů. Dominantní hodnotou zde bylo 14 paketů a 2316 bytů, což představovalo při sledovaných útocích zhruba 60% všech odpovědí na toky jdoucí směrem od útočníka.

4.4 Útok odepřením služby

Útok odepřením služby označuje široké spektrum různorodých útoků skrze počítačovou síť. Cílem těchto útoků je úplné znepřístupnění nebo omezení přístupu k určité službě. Většinou se jedná o webový, DNS nebo emailový server. Dále může být útok použit pro vynucení restartu serveru poté, co na něj byl nahrán útočníkem škodlivý kód. Tyto útoky je možné rozdělit dle jejich anatomie do dvou následujících skupin:

Útok hrubou silou V anglickém jazyce je tento typ útoku většinou nazýván *Flood*, záplava. Tyto útoky jsou postaveny na zaplavení cílového stroje velkým počtem dat, která je nutné zpracovat. Spoléhají na vyčerpání paměti nebo spotřebování procesorového času. Typickými představiteli jsou útok SYN flood, UDP flood a HTTP flood.

Využití chyby Tyto útoky se nazývají *Nuke*. Nejde zde o vyčerpání zdrojů oběti, ale útok spoléhá na zneužití chyby v implementaci služby. K shození počítače dojde pomocí zaslání jediného správně upraveného paketu. Nejznámějším příkladem je WinNuke, který využíval chyby v implementaci NetBIOS u Windows 95. Dnes se tyto útoky používají méně, protože je proti nim většina systémů chráněna.

Podle počtu útočníků a provedení je možné útoky rozdělit do tří následujících skupin:

DoS Jako Denial of Service je pojmenován útok, kdy útočí jeden útočník na jednu oběť. Dnes se téměř vůbec nepoužívá, protože jeden útočník není schopen zajistit převahu v rychlosti linky a hardwaru nad obětí tak, aby ji mohl zahltit požadavky.

DDoS Distributed Denial of Service je v dnešní době nejběžnějším druhem útoku. Do útoku bývají zapojeny počítače infikované malwarem a sdružené do botnetu. Vzhledem k tomu, že počty počítačů zapojených do botnetů se různí od desítek tisíc až po statisíce, jsou tyto útoky často velmi ničivé.

DRDoS Útok Distributed Reflected Denial of Service probíhá tak, že útočník na cíl útočí nepřímo pomocí podvržení zdrojové adresy požadavku, kam je umístěna adresa oběti. Poté je požadavek odeslán na co největší množství počítačů a ty poté svou odpovědí zahltí oběť.

4.4.1 HTTP flood

HTTP flood je jeden z nejjednodušších, ale při správném provedení i nejzákeřnější útoků [15]. Používán je proti webovým serverům a jeho cílem je znepřístupnit webové stránky běžící na tomto serveru. Principem útoku je zasílání běžných či nesmyslných GET požadavků na server. Tyto požadavky jsou navíc z hlediska firewallu legitimní a firewall tedy nemá důvod je filtrovat. Tím dochází k zahlcení serveru, který musí tyto nesmyslné požadavky zpracovávat a nezbývají mu tedy již systémové zdroje pro zpracování legitimních požadavků od uživatelů. Pro zdárné provedení útoku je potřeba mít k dispozici dostečně velké množství adres, ze kterých bude veden útok. Jedná se tedy o DDoS útok. Velmi často je tento útok implementován v botnetech [15].

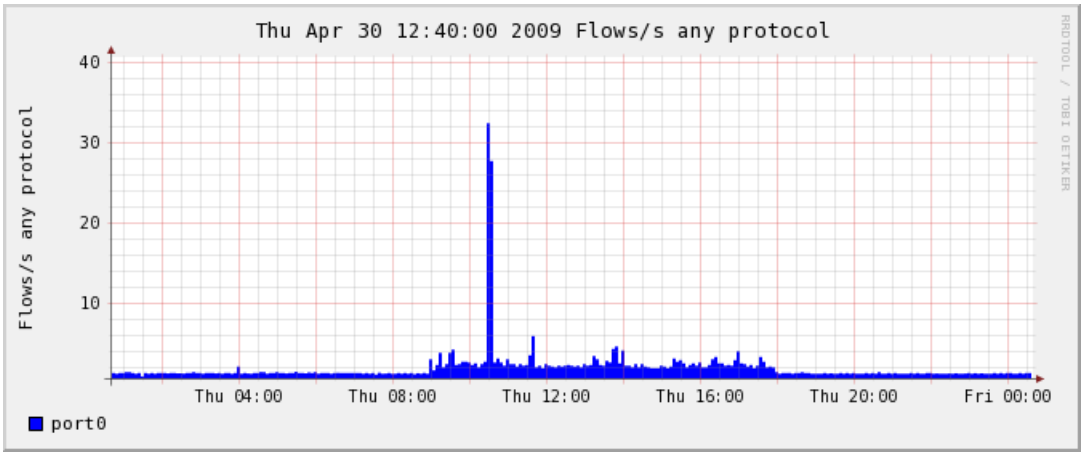
Simulovaný HTTP Flood útok

V rámci pokusů pro bakalářskou práci byl proveden simulovaný DDoS HTTP flood útok na server monitorovaný pomocí NetFlow dat. Útok byl simulován pomocí jednoduchého skriptu v jazyce Perl, který opakovaně v nekonečné smyčce odesílá na server požadavek „GET /“. Výsledek útoku na server je zachycen v grafu 4.1. Útok byl veden ze tří počítačů v celkové délce 5-7 minut. Za tuto dobu bylo uskutečněno celkem 16532 flow. Přeneseno bylo 82021 paketů s celkovou velikostí 8.3 MB.

Detekce útoku HTTP Flood útoku

HTTP flood je třeba hledat na portech, na kterých naslouchají webové servery ve sledované síti, typicky port 80 a šifrovaný port 443. Ostatní porty jsou při tomto typu útoku nezajímavé. Útok je charakteristický tím, že z útočících počítačů přichází mnoho spojení v krátkém časovém intervalu. Spojení se vyznačují malým počtem paketů a malým objemem přenesených dat a jsou si navzájem velmi podobná. Odpovědi na spojení jsou z hlediska počtu paketů a velikosti dat téměř stejné, opět s malým počtem přenesených dat a nízkým počtem paketů.

Při provedeném pokusu měly téměř všechny požadavky, tedy flow směřující od útočníků k oběti, velikost 280-400 bytů a počet paketů byl v rozmezí 5-6. V opačném směru měla flow velikost 700-800 bytů při pěti paketech 4.2.



Obrázek 4.1: V grafu je jasně viditelný okamžik, kdy došlo k útoku

	Duration	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp
10:34:38.995	0.002	147.229.XXX.aaa:58650	>	192.168.3.110:80	.AP.SF	0	5	336	2500	1.3 M	67
10:34:38.997	0.002	147.229.XXX.aaa:58651	>	192.168.3.110:80	.AP.SF	0	5	336	2500	1.3 M	67
10:34:39.275	0.002	147.229.XXX.bbb:36054	>	192.168.3.110:80	.AP.SF	0	5	280	2500	1.1 M	56
10:34:39.283	0.002	147.229.XXX.bbb:36057	>	192.168.3.110:80	.AP.SF	0	5	280	2500	1.1 M	56
10:34:39.285	0.002	147.229.XXX.bbb:36058	>	192.168.3.110:80	.AP.SF	0	5	280	2500	1.1 M	56
10:34:39.288	0.002	147.229.XXX.bbb:36059	>	192.168.3.110:80	.AP.SF	0	5	280	2500	1.1 M	56
10:34:39.302	0.002	147.229.XXX.bbb:36065	>	192.168.3.110:80	.AP.SF	0	5	280	2500	1.1 M	56
10:34:41.061	0.002	147.229.XXX.CCC:37709	>	192.168.3.110:80	.AP.SF	0	5	336	2500	1.3 M	67
10:34:41.063	0.002	147.229.XXX.CCC:37710	>	192.168.3.110:80	.AP.SF	0	5	336	2500	1.3 M	67
10:34:41.065	0.002	147.229.XXX.CCC:37711	>	192.168.3.110:80	.AP.SF	0	5	336	2500	1.3 M	67

Obrázek 4.2: Příklad flow směřujících od útočníků k cílovému serveru

Pro detekci HTTP flood útoku se jeví jako výhodnější použít flow směřující ze serveru k útočníkům 4.3, protože parametry flow v opačném směru se mohou měnit v závislosti na způsobu implementace útoku. Záleží zde na tom, jak složité dotazy dokáže nástroj použitý útočníkem skládat. Toky ze serveru jsou ovšem vždy velmi podobné, protože se jedná buď o odesílání stránek, nebo odesílání HTTP stavových kódů s doplňujícími informacemi. Detekční pravidlo tedy má následující tvar:

```
proto TCP and port 80 and packets 5 and (bytes > 700 and bytes < 800)
```

	Duration	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp
10:34:38.997	0.002	192.168.3.110:80	>	147.229.XXX.aaa:58651	.AP.SF	0	5	767	2500	2.9 M	153
10:34:39.277	0.002	192.168.3.110:80	>	147.229.XXX.bbb:36055	.AP.SF	0	5	711	2500	2.7 M	142
10:34:39.295	0.003	192.168.3.110:80	>	147.229.XXX.bbb:36062	.AP.SF	0	5	711	1666	1.8 M	142
10:34:39.307	0.003	192.168.3.110:80	>	147.229.XXX.bbb:36067	.AP.SF	0	5	711	1666	1.8 M	142
10:34:41.063	0.002	192.168.3.110:80	>	147.229.XXX.CCC:37710	.AP.SF	0	5	767	2500	2.9 M	153

Obrázek 4.3: Oddchozí toky ze serveru popsané pomocí výše zmíněného pravidla

4.5 Instant Messengery

Messengery, jak jsou programy pro online komunikaci mezi uživateli nazývány, představují pro datovou síť hrozbu z toho důvodu, že představují otevřený komunikační kanál, po

kterém se může šířit malware a warez ve všech podobách do jinak dobře zabezpečené sítě. Zde lze například zmínit rozšíření červů rodiny Stration [23], s alternativním názvem Warezov, který se šířil také pomocí odkazů odesílaných v síti ICQ. Po kliknutí na URL adresu obsaženou ve zprávě stáhl do počítače další malware. Tento červ byl rovněž odpovědný za rozesílání spamu. Pomocí messengerů také mohou unikat ven z vnitřní sítě citlivé informace v libovolném množství. V neposlední řadě není ideální, aby zaměstnanci trávili podstatnou část pracovní doby komunikací s přáteli pomocí Instant Messengerů.

4.5.1 Protokol OSCAR

Protokol OSCAR, celým názvem Open System for CommunicAtion in Realtime je protokol, který používá firma AOL pro své instant messengery ICQ a AIM. Jedná se o proprietální binární TCP protokol. Do nedávné doby byly všechny informace o protokolu dostupné pouze skrze reverzní inženýrství. Dne 5.března 2008 společnost AOL uveřejnila dokumentaci popisující některé části protokolu, aby ulehčila práci vývojářům. Dokumentace je dostupná na webových stránkách [7].

Na začátku komunikace se klient připojí na přihlašovací server společnosti AOL, kde se pokusí přihlásit ke svému účtu. Po úspěšném přihlášení je klient přihlašovacím serverem přepojen na komunikační server, přes který poté probíhá po celou dobu trvání jeho připojení komunikace s ostatními klienty sítě.

4.5.2 Protokol XMPP

Protokol XMPP je otevřený protokol na bázi XML sloužící pro instant messenging. Kromě použití v jabber serverech byl také použit firmou Google v jejich modifikaci Google Talk. XMPP je také používán pro vnitrofiremní komunikaci. Mezi výhody protokolu patří možnost použití šifrování a decentralizace. To znamená, že téměř kdokoliv si může spustit vlastní Jabber server a připadné blokování bude mnohem náročnejší, než u dříve uvedeného protokolu OSCAR.

Komunikace mezi klientem a serverem probíhá obdobně jako u protokolu OSCAR s tou výjimkou, že v komunikaci není přítomen žádný přihlašovací server nebo spíše přihlašovací server je i serverem komunikačním. Problém komunikace s uživateli na jiném serveru, jako důsledek decentralizace, probíhá na úrovni server-to-server a je pro klienta transparentní. To znamená, že například uživatel připojený přes jiný server může využívat služby na svém domovském serveru.

4.6 Detekce Instant Messengeru

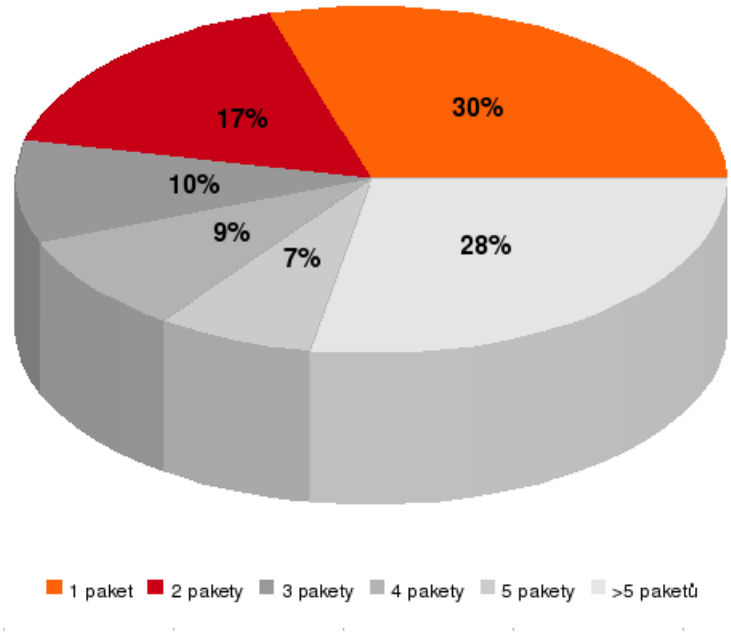
4.6.1 Obecná detekce Instant Messengeru

Výhodou detekce messengerů v takto obecném tvaru je, že není třeba tvořit detekční pravidla pro každý známý protokol zvlášť, ale že je možné odhalovat různé messengery pomocí stejné sady pravidel. Daní za takto obecný přístup je ovšem větší nepřesnost než v případě pravidel zacílených na specifický protokol. V tomto případě byly pozorovány protokoly OSCAR a XMPP, které jsou použity u messengerů ICQ, AIM, Google Talk a Jabber.

Obecná detekce vychází z vlastnosti společné pro všechny messengery jako takové. Touto společnou vlastností je zprostředkování komunikace mezi uživateli. Protože je třeba messenger pouze detekovat a ne sledovat jeho chování, je tedy možné ignorovat jeho sekundární

funkce, jako je například přenos souborů. Tyto funkce neposlouží při detekování přítomnosti messengeru v síti, protože se odehrávají pouze minimální množství času z komunikace mezi uživateli. Navíc by jejich zahrnutím do jediného detekčního pravidla došlo ke snížení úspěšnosti, protože přenos souborů vykazuje diametrálně odlišné charakteristiky než prostá komunikace mezi uživateli.

Vzhledem k tomu, že hledání Instant Messengerů probíhá v celém rozsahu portů a ip adres, není možné tyto údaje použít pro vytvoření detekčního pravidla. První věcí, na kterou je možno se spolehnout u komunikace messengerů, je fakt, že spojení mezi klientem a serverem, případně mezi klienty navzájem, zajišťuje TCP protokol. Z pozorování reálné komunikace vyplynulo, že 47% všech flow u sledovaných protokolů obsahuje méně než 3 pakety. Procentuální zastoupení flow podle počtu paketů v komunikaci je patrné v grafu 4.4.



Obrázek 4.4: Graf zobrazující počet paketů v jednotlivých flow v sledované síti

Dalším zjištěným faktem je, že 67% datových toků komunikace messengerů obsahuje v paketech příznaky ACK a PUSH. Nyní je třeba určit, jaký je průměrný počet bytů v paketech v jednotlivých flow komunikace. Vzhledem k tomu, že se jedná o obecnou detekci, je třeba zvolit tuto hodnotu velmi pečlivě, protože v případě příliš velkého rozsahu hodnot se bude zvyšovat počet chybně detekovaných flow. Po řadě pokusů byl rozsah ustanoven na hodnotách 200-400 bytů. Dále bylo vypozorováno, že hodnota *Počet bytů za sekundu* (bps) je u těchto flow v naprosté většině případů rovna nule. Pravidlo je zapsáno ve tvaru:

```
proto TCP and (flags AP and not flags RFUS) and (bpp > 200 and bpp < 400)
 and (packets < 3) and tos 0 and bps 0
```

Jak již bylo dříve uvedeno, je tato detekce relativně nepřesná, protože není zaměřena detailně na určitý protokol messengeru. To také představuje její největší přednost, protože

není třeba pravidlo upravovat, pokud budou stávající protokoly lehce poupraveny, což se děje například u protokolu OSCAR poměrně často. Pomocí tohoto pravidla byl také úspěšně detekován i jabber klient komunikující se serverem na portu 443, který je běžně určen pro HTTPS komunikaci, a proto je možné se přes něj připojit i skrze restriktivní firewall.

4.6.2 Detekce protokolu OSCAR

Pravidla pro detekci protokolu OSCAR byla sestavena na základě pozorování jeho charakteristických projevů v NetFlow datech. Cílem bylo naleznout často se opakující vzorek dat a umožnit tak rychlou detekci klienta v síti.

Po řadě pokusů bylo nalezeno první flow, které směřuje od klienta směrem k serveru. Ve flow jsou obsaženy dva pakety a *Počet bytů na paket* (bpp) je 40. Jediným nastaveným flagem je A (Ack) Pro detekci slouží následující pravidlo:

```
(flags A and not flags RPFUS) and bpp 40 and bps 0 and packets < 3
```

Druhým datovým tokem je tok směřující v opačném směru s následujícím zápisem:

```
(((flags AP and not flags RFUS) and (bytes 76 or bytes 75) and packets 1)
 or ((flags AR and not flags PFUS) and (bytes 40) and packets 1))
and (vystup pravidla 1)
```

4.6.3 Detekce protokolu XMPP

Detekce protokolu XMPP je stejně jako u předchozího protokolu založena na jeho pozorování a hledání pro něj chrakteristických flow. Pravidla pro tato flow byla otestována proti čtyřem serverům. Konkrétně se jednalo o tyto servery: Google Talk server na portu 5222, jabber server FIT VUT na portu 5223 a servery Jabbim na portech 5222 a 443. Vypozorovaná pravidla jsou následující:

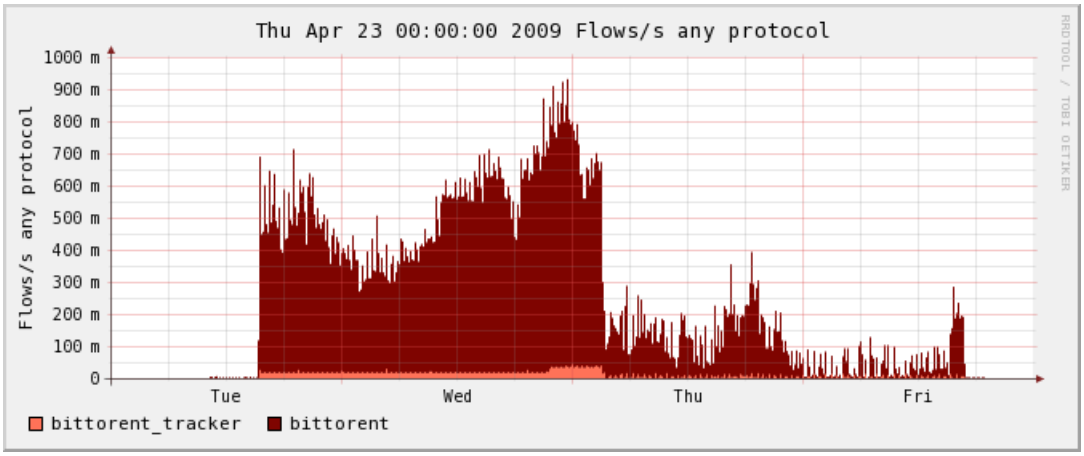
```
(bytes 77 or bytes 366) and (flags AP and not flags RFUS) and packets 1

(flags A and not flags RPFUS) and bpp 40 and bps 0 and packets < 3
and (vystup pravidla 1)
```

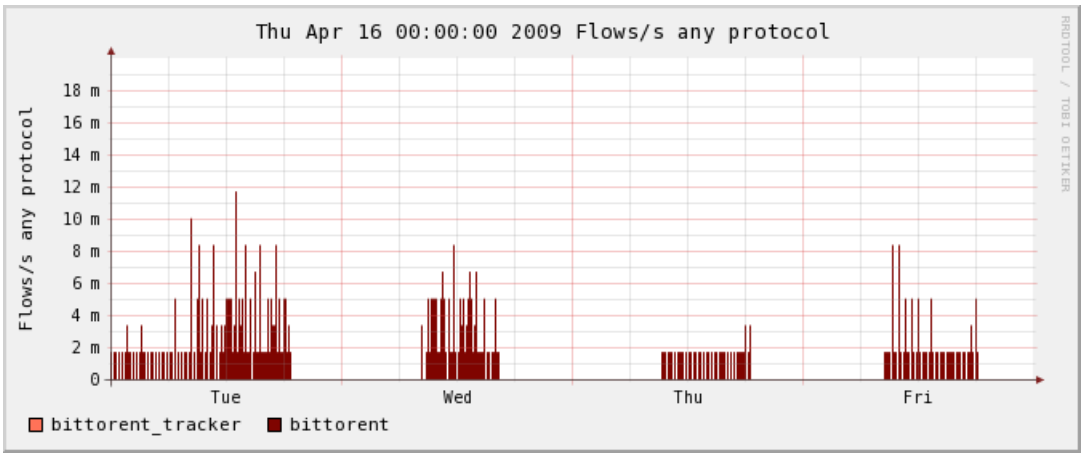
Není bez zajímavosti, že druhé detekční pravidlo je stejné jako pravidlo číslo jedna u detekce protokolu OSCAR. Toky se odlišují pouze směrem. Zde flow směřuje od serveru ke klientovi.

4.7 Sdílení dat pomocí protokolu BitTorrent

BitTorrent je protokol určený pro peer-to-peer (p2p) sdílení dat navržený Bramem Cohenem. Jeho první implementace byla vydána 2.července 2001. Od doby svého vydání se stal jedním z nejběžnějších protokolů pro sdílení souborů. Provoz protokolu BitTorrent představoval v roce 2004 celých 35% provozu v internetu [24]. Závažným problémem je, že BitTorrent klienti navazují za krátký časový okamžik velké množství spojení. Může jít i o 300-500 spojení za sekundu, což rychle zaplňuje překladové tabulky NAT na routerech. Rozdíl mezi sítí s jediným běžícím BitTorrent klientem a sítí bez BitTorrent provozu je pro ilustraci zachycen na obrázcích 4.5 a 4.6.



Obrázek 4.5: Síť s běžícím BitTorrent klientem



Obrázek 4.6: Síť za normálního stavu

Torrent Torrent označuje jak data stažitelná pomocí protokolu BitTorrent, tak soubor s příponou .torrent. Tento soubor obsahuje metadata všech souborů, které pomocí něj mohou být staženy. Konkrétně se jedná o názvy souborů, jejich velikosti a také kontrolní součty všech částí torrentu. Dále je v něm obsažena adresa Trackeru.

Peer Takto je označována jedna instance BitTorrent klienta v síti internet. Navazuje spojení s ostatními peery a pokouší se zkompletovat stahované soubory. Při zjednodušení terminologie lze považovat slovo *peer* za synonymum pro *klienta*.

Tracker Tracker je označení pro server, který udržuje pohromadě různé torrent soubory a poskytuje klientům informace o ostatních klientech spolupracujících na kompletaci konkrétního torrentu. Provoz mezi klientem a trackerem obvykle probíhá pomocí HTTP/HTTPS komunikace. Mezi nejznámější trackery patří Isohunt, The Pirate Bay a Mininova. Existuje ovšem obrovské množství dalších trackerů. Některé z nich jsou veřejné a některé soukromé (pouze na pozvánku).

Swarm Swarm je označení pro skupinu peerů sdílejících data patřící do stejného torrentu.

Sdílení dat postavené nad protokolem BitTorrent je pomocí běžných technik obtížně blokovatelné. Tradiční techniky spoléhají na blokování rozsahu portů, což je vzhledem k tomu, že program může běžet nad libovolným portem, neúčinné. Dalším běžným způsobem je blokování spojení s Trackerem a tím znemožnění vyhledání protějšků pro stahování a následného stažení souboru. Vzhledem k tomu, že databáze sloužící k blokování tohoto provozu nebude pravděpodobně nikdy znát všechny Trackery v internetu, je úspěšnost této metody rovněž diskutabilní.

Jako pokusný objekt byl použit počítač s běžícím programem rTorrent [18]. Na tento počítač byla stažena linuxová distribuce a poté byla po čtyři dny sdílena. Takto dlouhá doba byla určena proto, že se nejedná o příliš často stahovaný obsah a tak bylo třeba počkat na klienty hledající tato konkrétní data. Výhodou přístupu bylo, že v době kdy se tvořila monitorovací pravidla, byla zjednodušena kontrola provozu, protože byla známa ip adresa stroje, na kterém BitTorrent klient běží a všechny další okolnosti byly pod kontrolou.

Při vytvoření detekce bitTorrent protokolu byla výchozím materiálem specifikace protokolu uveřejněná na webových stránkách [6] a [25]. Z hlediska NetFlow dat není možné odlišit kvůli absenci obsahu paketů, stahování dat z Trackeru od ostatní HTTP/HTTPS komunikace. Pro úspěšnou detekci tedy bylo nutné naleznout vzorek flow charakteristik samotného sdílení dat a komunikace mezi klienty.

První zjištěný poznatek je, že pakety vysílané při komunikaci bitTorrent klienty mají nastaveny hodnotu Type of Service na 8, což znamená maximální propustnost [2].

Dalším poznatkem je, že se klienti po stažení torrent souboru z trackeru a dotazu ohledně klientů na tracker pokouší navázat spojení s ostatními aktivními klienty poskytujícím stejná data, aby mohli spolupracovat na kompletaci souboru. V NetFlow datech se tento pokus o spojení projeví jako série stejných nebo velmi podobných toků v krátkém časovém okamžiku. Tuto komunikaci je možné zachytit pravidlem:

```
((flags S and not flags ARPFU) or (flags A and not flags RPFUS))
 and (bpp > 50 and bpp < 61) and packets < 4 and tos 8
```

Důležitým výstupem tohoto pravidla jsou ip adresy a čísla portů mezi počítači, kde panuje podezření na běžícího BitTorrent klienta. Toto podezření se úměrně zvyšuje s počtem nalezených flow se stejnou charakteristikou. Pokud je v rámci jednoho souboru s NetFlow daty zachyceno více flow z určitého zdroje s výše uvedenými charakteristikami, pak se jedná s velkou pravděpodobností o stroj s běžícím BitTorrent klientem. Nyní je třeba podezření potvrdit pomocí vyhledání proběhlého přenosu dat.

Zde je možné z komunikace mezi klienty detekovat dva druhy výměny dat. První je požadavek jednoho klienta druhému na odeslání seznamu bloků, které má k dispozici pro výměnu. Velikost požadavku v bytech je podle specifikace protokolu definován jako 16-32 kilobytů [25]. Druhá detekovatelná událost je vlastní výměna dat mezi BitTorrent klienty. Dle specifikace je maximální velikost bloku 1MB [25]. Nejlepší je dle [25] nastavit velikost bloku na 512kB nebo méně.

Obě výše uvedené činnosti je možné vyhledat pomocí následujícího pravidla:

```
((bytes > 15k and bytes < 33k) or ((bytes > 255k and bytes < 1m)
 and bpp > 1000)) and proto TCP and (vystup pravidla 1)
```

Kapitola 5

Závěr

Na základě seznámení se s technologií NetFlow a po prostudovanáním různých hrozeb v datové síti a jejich projevů v NetFlow datech, byla navržena detekční pravidla pro odhalování výše uvedených hrozeb a nežádoucího provozu. Důležitá byla přitom co největši možná obecnost hledaných řešení, což se podařilo splnit. Například tam, kde není explicitně požadováno sledování určitého portu, jsou pravidla na portech naprosto nezávislá. Dále bylo pro použití s pravidly navrženo vyhledávání pomocí dvou pravidel. Při tomto vyhledání jsou použity informace získané prvním pravidlem upřesněna pomocí pravidla číslo dvě.

V rámci práce bylo provedeno několik testů s počítačem sloužícím jako pokusný objekt a umístěným v síti monitorované pomocí NetFlow. Jednalo se hlavně o sledování chování BitTorrent klienta a nalezení způsobu pro jeho účinné odhalení. V menší míře byl počítač použit pro otestování odhalení jabber klienta komunikujícího na portu pro https komunikaci. Dalším pokusem byl po domluvě se správcem sítě vedený simulovaný DDoS útok.

Nejdůležitějším bodem této práce bylo objevení principů a vzorů popisovaných hrozeb. Po té, co byly objeveny charakteristické toky v popisovaných hrozbach, bylo mozné tyto hrozby efektivně odhalit. Nyní je možné zjištěné výsledky zapracovat do libovolného systému pro detekci hrozeb v datové síti pomocí NetFlow dat a umožnit tak jejich odhalení.

Ochrana datové sítě se zdá být velmi perspektivní, zvláště na vysokorychlostních linkách internetových poskytovatelů. Zajímavá také může být pro univerzitní sítě a pro sítě velkých firem. V obou případech jde o správu rozsáhlých sítí, která by byla pomocí konvenčních prostředků poněkud obtížná. V blízké budoucnosti lze očekávat zvýšenou potřebu monitorování vysokorychlostních sítí, a proto lze předpokládat rozmach technologie NetFlow a jí obdobných řešení schopných pracovat na vysokorychlostních linkách.

Literatura

[1] Alkharobi, T.: Firewalls. Květen 2007, [cit. 2009-4-27].
URL http://ocw.kfupm.edu.sa/user062/CSE55101/firewall.pdf

[2] Almquist, P.: Type of Service in the Internet Protocol Suite. Červenec 1992, [cit. 2009-5-14].
URL http://rfc.sunsite.dk/rfc/rfc1349.html

[3] autorů, K.: Transmission Control Protocol. Srpen 1981, [cit. 2009-4-27].
URL http://www.ietf.org/rfc/rfc793.txt

[4] Bowers, D.: The State of Spam A Monthly Report–February 2009. Technická zpráva, Symantec, Březen 2009.

[5] Claise, B.: Cisco Systems NetFlow Services Export Version 9. Říjen 2004, [cit. 2009-4-27].
URL http://www.ietf.org/rfc/rfc3954.txt

[6] Cohen, B.: The BitTorrent Protocol Specification. Leden 2008, [cit. 2009-5-10].
URL http://www.bittorrent.org/beps/bep_0003.html

[7] Developer Network: OSCAR Protocol — dev.aol.com. Březen 2008, [cit. 2009-5-14].
URL http://dev.aol.com/aim/oscar/

[8] Fyodor: *Manuálová stránka programu Nmap*. insecure.org, Září 2007, dostupno také na http://nmap.org/book/man.html.

[9] Houser, P.: Jaký botnet nejvíc spamuje? Duben 2009, [cit. 2009-4-28].
URL www.securityworld.cz/securityworld/Jaky-botnet-nejvic-spamuje-1606

[10] Kasprzak, J.: Spam včera, dnes, a bohužel asi i zítra. [cit. 2009-4-28].
URL http://www.fi.muni.cz/~kas/papers/spam.pdf

[11] Žádník Martin: NetFlow. Stránky předmětu ISA, [cit. 2009-4-27].

[12] Quittek, J.; Zseby, T.; Claise, B.; aj.: Requirements for IP Flow Information Export (IPFIX). Říjen 2004, [cit. 2009-4-27].
URL http://www.ietf.org/rfc/rfc3917.txt

[13] Rehak, M.; Celeda, P.; Pechoucek, M.; aj.: CAMNEP: Multistage Collective Network Behavior Analysis System with Hardware Accelerated NetFlow Probes. [cit. 2009-5-13].
URL http://www.cert.org/flocon/2009/presentations/Rehak_Camnep.pdf

[14] Thomas, T. M.: *Zabezpečení počítačových sítí bez předchozích znalostí*. Computer Press, 2004, ISBN 80-251-0417-6.

[15] Turek, R.: Botnety.
URL http://blog.synopsi.com/2008-04-27/botnety#ddos

[16] Vykopal, J.; Plesnik, T.; Minarik, P.: Validation of the Network-based Dictionary Attack Detection. In *Security and Protection of Information 2009*, 2009, ISBN 987-80-7231-641-0, s. 128–136.

[17] WWW stránky: How to Choose A Firewall That is Right for Your Needs. [cit. 2009-4-27].
URL www.apluskb.com/scripts/How_to_choose_a_Firewall_answer187.html

[18] WWW stránky: The libTorrent and rTorrent Project.
URL http://libtorrent.rakshasa.no/

[19] WWW stránky: NetFlow Services Solutions Guide - Cisco Systems. [cit. 2009-5-15].
URL http://www.cisco.com/en/US/products/sw/netmgtsw/ps1964/products_implementation_design_guide09186a00800d6a11.html

[20] WWW stránky: Stránka programu NfDump.
URL http://netflow.caligare.com/applications.htm

[21] WWW stránky: Stránka programu NfDump.
URL http://sourceforge.net/projects/nfdump/

[22] WWW stránky: Stránka programu NfSen.
URL http://nfsen.sourceforge.net/

[23] WWW stránky: VIRY.CZ: Win32/Stration.
URL http://www.viry.cz/go.php?p=viry&t=popis&id=218

[24] WWW stránky: Bit Torrent: 35% of all Traffic. 2004, [cit. 2009-5-14].
URL http://www.dslreports.com/shownews/56403/

[25] WWW stránky: Bittorrent Protocol Specification v1.0. Březen 2009, [cit. 2009-5-5].
URL http://wiki.theory.org/BitTorrentSpecification

[26] WWW stránky: IDS a IPS systémy. [cit. 2009-4-28].
URL http://www.invea.cz/cs/solutions/ids-ips

Dodatek A

Obsah CD

- Zdrojové soubory této práce.
- Testovací data.
- Demonstrační skripty.