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

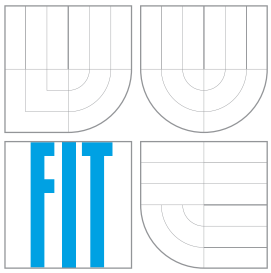
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETR SEDLÁŘ

BRNO 2010

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

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETR SEDLÁŘ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ TOBOLA

BRNO 2010

Abstrakt

Dokument poskytuje informace o technologii Cisco NetFlow a možnostech jejího využití pro ochranu počítačových sítí před různými druhy útoků. Součástí dokumentu je zmapování běžných bezpečnostních hrozeb z hlediska jejich detekovatelnosti na síťové a transportní vrstvě. Pro vybrané hrozby jsou určeny charakteristické vlastnosti, kterými se projevují v NetFlow datech a na jejich základě je navržena a implementována aplikace, která tyto hrozby detekuje.

Abstract

This document provides information about Cisco NetFlow technology and its usage to protect networks from different types of attacks. Part of the document is a summary of common security risks in term of their detection on network and transport layer. There are specified characteristics of NetFlow data containing samples of security risks. On the basis of these characteristics, an application for detection these risks is designed and implemented.

Klíčová slova

NetFlow, IPFIX, Bezpečnostní hrozby, Spam, IP telefonie, Peer-to-peer sítě, SQL

Keywords

NetFlow, IPFIX, Security threats, Spam, Voice over IP, Peer-to-peer networks, SQL

Citace

Petr Sedlář: Ochrana datové sítě s využitím NetFlow dat, diplomová práce, Brno, FIT VUT v Brně, 2010

Ochrana datové sítě s využitím NetFlow dat

Prohlášení

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

.....
Petr Sedlář
26. května 2010

Obsah

Kapitola 1

Úvod

S růstem naší závislosti na počítačových sítích rostou požadavky na jejich bezpečnost a spolehlivost. Tyto dva požadavky spolu úzce souvisí: Pro koncového uživatele, který potřebuje přístup ke konkrétním datům nebo síťové službě, je většinou nepodstatné, jestli je nedostupnost způsobena nějakou příčinou z oblasti spolehlivosti, například selháním hardware nebo třeba operačního systému nebo zda je to důsledek úspěšně provedého DoS (Denial of Service - znepřístupnění služby) útoku a patří tedy do oblasti bezpečnosti. Je proto pochopitelné, že oblasti bezpečnosti a spolehlivosti počítačových sítí bývá věnována stále větší pozornost.

Řešení bezpečnosti a spolehlivosti sítě není možné bez dobrého přehledu o tom, co se v dané síti děje. Pro efektivní detekování vnějších i vnitřních útoků na počítačovou síť a zejména na síťové servery, ale i pro potřeby hledání problémů s přetížením datových linek nebo optimalizaci sítě, vzniká potřeba mít k dispozici detailní informace o datových tocích, které na těchto sítích probíhají. K získání informací o datových tocích v různých místech sítě lze uspěšně využít technologii NetFlow společnosti Cisco.

Cílem této diplomové práce je demonstrace možností využití NetFlow dat v oblasti zabezpečení počítačové sítě. Práce se zaměřuje především na využitelnost této technologie k detekci síťových útoků a detekci bezpečnostních hrozeb. Postupy detekce navržené v tomto textu byly implementovány do prototypové aplikace, která byla následně otestována na různých vzorcích NetFlow dat.

V úvodu práce je čtenář detailně seznámen s technologií Cisco NetFlow. Popsána je arichitektura systému, funkce exporéru a kolektoru, uveden je přehled dostupných verzí protokolu. Detailně jsou zmíněny dvě nejrozšířenější verze, jejich možnosti a omezení pro zpracování informací o datových tocích v síti. Krátce představeny jsou i alternativy k NetFlow.

Následující kapitola se zabývá zmapováním běžných útoků proti počítačovým sítím z pohledu jejich možné detekovatelnosti na síťové a transportní vrstvě síťové architektury, tedy těch vrstev, se kterými technologie NetFlow primárně pracuje.

Stěžejní částí této práce je kapitola zaměřená na detekovatelnost hrozeb s využitím NetFlow statistik. Zde jsou pro jednotlivé hrozby analyzovány NetFlow data, ve kterých se nacházejí vzory síťového provozu odpovídajícího těmto hrozbám a na základě nalezených charakteristických vlastností, kterými se ve statistikách projevují, jsou navrhovány postupy jejich detekce. Pro navržené testy jsou doporučeny jejich konkrétní parametry a určeny faktory, které mohou správnost detekce negativně ovlivnit.

Čtvrtá kapitola se zaměřuje na popis implementace navržené demonstrační aplikace pro detekci vybraných hrozeb. Čtenář je seznámen se základním konceptem aplikace, předsta-

veny jsou technologie které byly při implementaci využity a zdůvodněno je jejich využití v řešeném projektu. Diskutovány jsou možnosti a omezení vytvořené aplikace.

Poslední část přináší souhrn informací z testování aplikace na různých vzorcích NetFlow dat. Popsány jsou případy, ve kterých detekce funguje lépe a ve kterých hůře a případy, kdy dochází k nesprávné detekci jsou zdůvodněny. Rovněž zde jsou diskutovány možnosti dalšího rozšíření projektu.

Tato diplomová práce navazuje na semestrální projekt řešený v průběhu zimního semestru 2009. V rámci tohoto projektu byl vytvořený text o technologii NetFlow, který z velké části odpovídá druhé kapitole této diplomové práce a dále text mapující bezpoečnostní hrozby z pohledu jejich detekovatelnosti na síťové a transportní vrstvě, který tvoří s některými úpravami třetí kapitolu této práce.

Kapitola 2

Technologie NetFlow

2.1 Úvod do NetFlow

V rámci správy počítačových sítí vznikla potřeba sledovat u aktivních prvků provoz v síti na vyšši urovni, než pouze objem přenesených dat. Zejména pro účely účtování dat, bezpečnostní analýzy, monitorování síťového provozu, optimalizace sítě nebo třeba zjišťování informací o struktuře přenášených dat pro marketingové účely. Zde všude je důležitá možnost přístupu k informacím o tom, které zařízení komunikuje s jakým, kdy, jak moc a jakým stylem.

Na tyto potřeby reagovala společnost Cisco vývojem protokolu NetFlow. Jde o proprietární protokol, který umožnuje sběr statistických informací o datových tocích procházejících směrovačem. Jeho hlavním účelem je monitorování síťového provozu na základě IP toků, které poskytuje administrátorům i manažerům podrobný pohled do provozu na jejich síti v reálném čase. Proto tvoří důležitou a nepostradatelnou součást zabezpečení každé počítačové sítě a je užitečný pro ISP (Internet Service Providers - poskytovatelé připojení), kteří na základě NetFlow statistik mohou svým zákazníkům účtovat ceny služeb v závislosti na množství přenesených dat. S pomocí NetFlow statistik lze odhalovat vnější i vnitřní incidenty, úzká místa v síti, dominantní zdroje provozu, efektivněji plánovat budoucí rozvoj sítě, sledovat, kdo komunikoval s kým, jak dlouho a s pomocí kterého protokolu [7].

Protokol NetFlow je v současné době dostupný již ve své deváté verzi, která je specifikována v RFC 3954 - Cisco Systems NetFlow Services Export Version 9.

2.2 Architektura

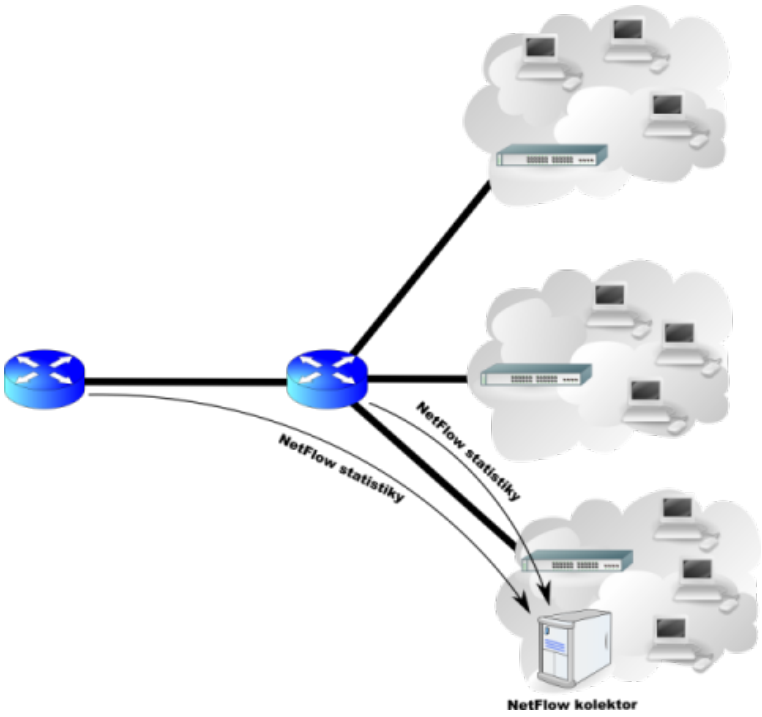
Základní části NetFlow architektury jsou NetFlow exportér a NetFlow kolektor.

2.2.1 Exportér

Exportér bývá nejčastěji součástí ativního zařízení typu router nebo switch, ale může se jednat i o samostatné zařízení v podobě autonomní NetFlow sondy připojené pomocí rozbočovače ke sledované lince. NetFlow exportér sleduje komunikaci na zvoleném portu aktivního prvku, kde analyzuje procházející pakety. Identifikuje z nich jednotlivé datové toky, zaznamenává pro ně počet přenesených bajtů a tuto infomaci po ukončení spojení přenese směrem ke kolektoru.

2.2.2 Kolektor

Kolektor získává informace o datových tocích z jednoho nebo i více exportérů a tyto ukládá do databáze. Takto uchovaná data mohou být následně analyzovány a využity pro detekci problémů na síti a jejich odstraňování, nebo k tvorbě různých přehledů, většinou v podobě tabulek a grafů, které umožňují jednoduše analyzovat monitorovaný provoz i běžnému uživateli.



Obrázek 2.1: Získávání dat z NetFlow exportérů [7]

Komunikace z exportéru na kolektor probíhá s využitím protokolu UDP, může tak dojít ke ztrátě paketu. Tuto ztrátu je možné detekovat podle sekvenčního čísla v hlavičce NetFlow záznamu, není ale možné iniciovat opakované poslání datagramu, protože kolektor zpracovaný záznam po odeslání ihned zahazuje. Do budoucna se pro přenos exportér kolektor počítá s využitím protokolu SCTP.

2.3 Datový tok

Datový tok je základní jednotkou sledovanou v NetFlow. Datový tok (flow) je definován jako jednosměrná sekvence paketů, která sdílí pětici vlastností:

- Zdrojová IP adresa
- Cílová IP adresa
- Zdrojový TCP/UDP port
- Cílový TCP/UDP port

- Použitý komunikační protokol (TCP/UDP)

V některých případech může být toto rozlišení ještě zjemněno pomocí dalších vlastností:

- číslo vstupního rozhraní aktivního prvku
- nastavená hodnota Type of Service (ToS) v hlavičce IP paketu

Obrázek 2.2: Identifikace datového toku

Exportér pro každý takto určený datový tok zaznamenává čas jeho vzniku a ukončení, množství přenesených bajtů a paketů a podle použité verze NetFlow protokolu i některé další informace. Po nasbírání určitého počtu ukončených toků je exportér přepošle na kolektor a svoji kopii těchto záznamů uvolní z paměti.

K ukončení datového toku a jeho následnému odeslání na kolektor dochází z pohledu exportéru v těchto případech:

- Po ukončení TCP spojení je zřejmě nejčastější případ. Datový tok je ukončen poté, co je v rámci tohoto toku zaznamenén průchod paketu s nastaveným příznakem FIN případně RST.
- U dlouhotrvajících toků po vypršení pevného časového intervalu. Zejména v případě, kdy jsou NetFlow záznamy využívané pro bezpečnostní analýzu provozu, není přijatelné, aby informace o dlouhodobém spojení byla odeslána na kolektor až po jeho ukončeni (například po několika dnech od svého vzniku). Proto je záznam po vypršení nasteveného časového limitu ukončen a odeslán i v případě, že sledované datové spojení stále probíhá.
- Po vypršení časového intervalu od posledního paketu v rámci toku. Tímto způsobem jsou ukončovány především UDP datové toky, ale i TCP spojení, která nebyla řádně ukončena (např. v případě ztráty spojení mezi stanicemi).
- V případě zaplnění paměti nebo při hrozbě přetečení čítačů.

2.4 Verze NetFlow

Přehled jednotlivých verzí protokolu NetFlow:

Verze	Poznámka
v1	první uvedená verze datového formátu NetFlow
v5	oproti první verzi přidává informace o čísle autonomního systému BGP a sekvenční číslo datového tokou
v6	informace o zapouzdření
v7	informace o přepínání
v8	agregace více toků do jednoho
v9	aktuální verze protokolu, přináší nový formát, který je rozšiřitelný o nové položky a typy záznamů díky šablonovému designu. Specifikován v RFC 3954 (2004) [5]

Verze 2 až 4 nebyly nikdy veřejně uvedeny [1]. Dnes jsou prakticky používány pouze verze 5 a 9, v dalším textu budou proto uvažovány pouze tyto dvě verze.

2.4.1 NetFlow v5

Datový formát NetFlow v5 má pevně dané datové položky. Každý záznam obsahuje: číslo verze, sekvenční číslo, vstupní a výstupní rozhraní, čas počátku a konce datového toku, počet bajtů a paketů toku, zdrojová a cílová IP adresa, zdrojový a cílový port, IP protokol, TCP příznaky, IP Type of Service a některé další. To s sebou nese některé nevýhody: jednak je nutné v exportéru zpracovávat a přeposílat na kolektor i informace, které nemusí být pro danou aplikaci nijak užitečné, a naopak neumožňuje získavat informace, které by užitečné byly, ale NetFlow v5 je nepodporuje (např. IPv6, MPLS).

Každá zpráva protokolu NetFlow, zasílaná v paketu transportního protokolu UDP z exportéru na kolektor, se skládá z hlavičky následované jedním až třiceti záznamy o uskutečněných datových tocích. Hlavička zprávy má následující formát:

Bajty	Obsah	Popis
0-1	version	Verze NetFlow protokolu
2-3	count	Počet toků přenášený v tomto paketu (1-30)
4-7	sys_uptime	Čás od startu exportéru (v milisekundách)
8-11	unix_secs	Unixový čas exportéru (v sekundách)
12-15	unix_nsecs	Zbytkové nanosekundy k unixovému času exportéru
16-19	flow_sequence	Počítadlo toků celkem zpracovaných exportérem
20	engine_type	Typ zařízení exportéru
21	engine_id	Číslo slotu exportéru
22-23	sampling_interval	Vzorkovací interval

Formát hlavičky NetFlow v5 paketu

Formát záznamů o konkrétním datovém toku, kterých může za touto hlavičkou paketu následovat až 30, má v NetFlow verze 5 tento formát:

Bajty	Obsah	Popis
0-3	srcaddr	Zdrojová IP adresa
4-7	dstaddr	Cílová IP adresa
8-11	nexthop	IP adresa příštího routeru
12-13	input	SNMP index vstupního rozhraní
14-15	output	SNMP index výstupního rozhraní
16-19	dPkts	Celkový počet paketů v toku
20-23	dOctets	Celkový počet přenesených bytů (až po hlavičku 3.vrstvy včetně)
24-27	first	Uptime čas exportéru při zažátku toku
28-31	last	Uptime čas exportéru při průchodu posledního paketu v toku
32-33	srcport	TCP/UDP zdrojový port
34-35	dstport	TCP/UDP cílový port
36	pad1	Nepoužito (zarovnání)
37	tcp_flags	Logický součet TCP přízanků ze všech paketů toku
38	prot	Číslo přenášeného protokolu
39	tos	Nastavený ToS (type of service) v hlavičce IP paketu
40-41	src_as	Číslo autonomního systému, ze kterého spojení přichází
42-43	dst_as	Číslo autonomního systému, do kterého spojení směřuje
44	src_mask	Síťová maska prefixu zdorjových adres
45	dst_mask	Síťová maska prefixu cílových adres
46-47	pad2	Nepoužito (zarovnání)

Formát záznamu datového toku v NetFlow v5

2.4.2 NetFlow v9

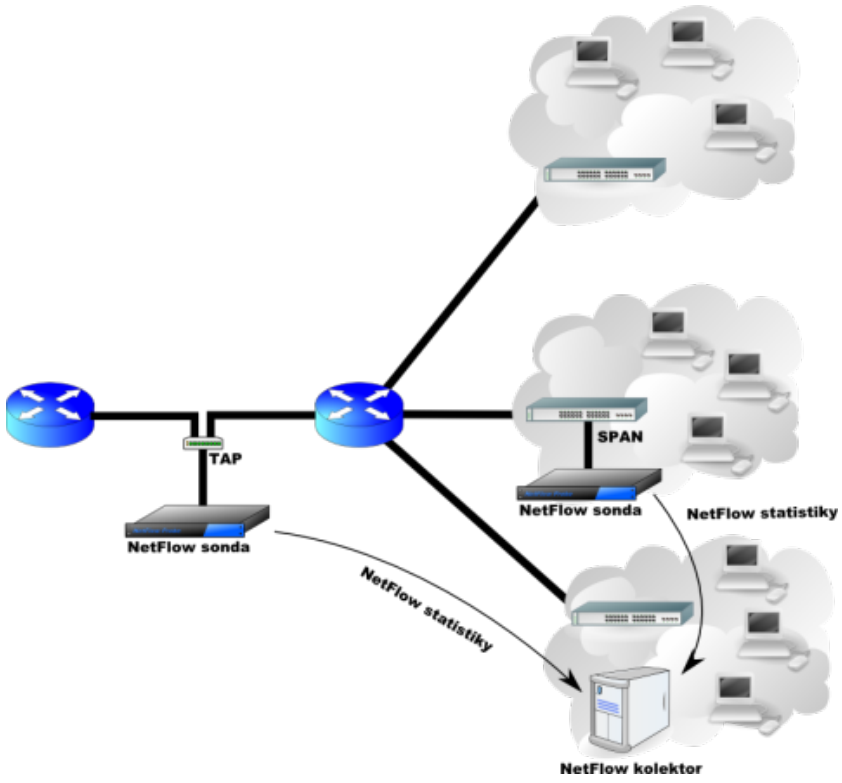
Přichází s novým systémem šablon, kde je možné uživatelsky definovat sledovaná data, která, exportér zpracovává a zaznamenává. Podporuje také agregované datové toky. Ty

umožňují v jednom datovém toku sledovat například celou jednu podsíť, namísto rozlišení toků pro jednotlivé adresy z této podsítě. Mezi nejvýznamnější vlastnosti, které NetFlow v9 umožňuje nově zpracovávat, patří především: Multiprotocol Label Switching labels (MPLS), IPv6 adresy a porty, počet agregovaných toků a jejich masky [2].

2.5 Výkonnostní problémy

Identifikace toků je velmi náročná na paměť a na procesor, proto je běžně NetFlow povolováno jen na vstupu u vybraných rozhraní. I přesto náročnost zpracování NetFlow statistik pro větší datové toky, především na gigabitových nebo desetigigabitových linkách, převyšuje možnosti běžných směrovačů. Proto se zavádí vzorkování, při kterém se pro potřeby tvorby statistik zpracovávají pouze vybrané pakety, např. 1 z N, nebo jeden paket za určitý časový interval. Tento režim bývá označován jako vzorkovaný NetFlow (Sampled NetFlow). Vzorkování snižuje přesnost měření, pro provoz velké částí aplikací nad NetFlow daty to ale není příliš významné.

Dalším používaným řešením je použití samostatné pasivní NetFlow sondy, která se připojí ke sledované datové lince pomocí optického/metalického robočovače nebo na samostatný port aktivního prvku, který je nastaven v režimu zrcadlení jiného portu, pro který chceme zpracovávat data. Sondou data neprochází a proto je celý její výpočetní výkon dostupný pro zpracování NetFlow statistik.



Obrázek 2.3: Příklad použití NetFlow sondy [7]

2.6 NetFlow a ostatní výrobci

Kromě Cisca, které má funkčnost NetFlow kolektoru implementovanou u většiny směrovačů, nabízí tuto funkcí i ostatní výrobci aktivních prvků, často s drobými úpravami a pod jiným názvem:

- Huawei Technology - NetStream
- Juniper - CFlow (NetFlow v5 a v8)
- Mikrotik - Traffic Flow

Dostupné jsou také různé implementace na aplikační úroveňi pro OS Linux, FreeBSD a další (nProbe, fprobe, softflowd) nebo samostatné hardwarově akcelerované NetFlow sondy (FlowMon Probe od INVEA-TECH).

2.7 IPFIX

IPFIX (IP Flow Information Export) je připravovaný IETF standard, vycházející z NetFlow v9 (někdy je proto označován jako NetFlow v10). Požadavky na tento nový formát byly specifikovány v RFC 3917 z roku 2004. Je založen na NetFlow v9, ke kterému přidává další rozšíření. Nově přidává například podporu protokolu TCP pro komunikaci mezi exportérem a kolektorem. Snaží se o další zobecnění, standardizaci a vytvoření společného exportního protokolu. Přestože stále ještě nebyl definitivně schválen, začínají se objevovat jeho implementace v zařízeních různých výrobců.

Kapitola 3

Bezpečnostní hrozby

Jak je zřejmé z popisu technologie NetFlow v předchozí kapitole, všechny zpracovávané údaje o datovém provozu v síti se týkají síťové a transportní vrstvy OSI modelu síťové architektury. Proto bude v následujícím přehledu síťových bezpečnostních hrozeb brána do úvahy především jejich detekovatelnost na těchto dvou úrovních.

3.1 Denial of Service (DoS)

DoS je druh útoku, který nemá za úkol odcizit data nebo získat kontrolu nad napadnutým systémem, ale útočník jej používá k podpoře svých plánů. Princip tkví v zahlcení serveru daty. Server pak pod tíhou dat nevydrží a zkolabuje. Silnější variantou je pak DDoS (distribuovaný DoS), který ke své práci využívá více stroju, jejichž síla se spojí. Tím lze vytvořit datový tok, který i o několik řádů převyšuje běžný provoz. Takovému útoku se odolává velmi špatně. K posílení ničivé síly se navíc většinou používají buďto vadné pakety, nesmyslné dotazy, nebo otevírání spojení, která poté nejsou uzavírána. Tím se zátěž ještě znásobí [3].

Detekce DoS a především DDoS útoku je možná sledováním parametrů síťového provozu jako je počet spojení z jedné IP adresy, celkový počet spojení na server, celkový datový tok. Překročení určitého limitu lze vyhodnotit jako DoS/DDoS útok a podniknout odpovídající opatření. Lze využít také analýzu statistických infomací o aktuálním datovém toku a srovávat je s dlouhodobými charakteristikami pro daný stroj. Ne vždy sice musí být razantní nárůst v některém ze sledovaných parametrů způsoben DoS nebo DDoS útokem, ale i v tom případě si velmi výrazná odchylka od běžného provozu pravděpodobně zaslouží pozornost správce a tak vyvolaná falešná pozitivní detekce DoS útoku nemusí být zásadní vadou.

3.2 Spam

Spam je nevyžádané masově šířené sdělení (nejčastěji reklamní) rozesílané internetem především prostřednictvím e-mailu. Z hlediska síťové bezpečnosti není ani tak závažný problém s množstvím nevyžádaných zpráv, končících ve schránkách uživatelů sítě, jako spíš případ kdy se spammer nachází v naší síti, nejčastěji v podopbě napadeného počítače. Většina spamu je totiž rozesílána distribuovaně z počítačů napadených počítačovým virem nebo červem. Vir nebo červ často na počítači otevírá tzv. zadní vrátka (backdoor), která umožňují útočníkovi počítač dálkově ovládat a zneužít jej mj. pro rozesílání spamu. Rozesílací robot i databáze adres může být na napadený počítač zaslána ad hoc, rozesílání nemusí probíhat neustále [8].

Jedním ze závažných důsledků může být zařazení IP adres používaných sítí, ve které se vyskytují spamující počítače, na veřejné blacklisty a s tím spojený následný problém s doručováním legitimních e-mailových zpráv. Problém, zejména při větším množství takto aktivních stanic v síti, může být i značné zvýšení zátěže komunikačních linek a aktivních prvků. Nezanedbatelný je i fakt, že takto napadený počítač ovládaný útočníkem, může být lehce použit i k jiným, často i závažnějším účelům, než rozesílání spamu. Například pro různé druhy útoků proti síti, ve které se nachází.

Detekce spamujících stanic v síti na síťové a transportní vrstvě je možná díky značně odlišným charakteristickým znakům legitimní a nežádoucí e-mailové komunikace. Zatímco spam má obvykle podobu velkého množství navazovaných TCP spojení na port 25 na velké množství různých hostitelů v krátkém čase, legitimní komunikace má charakter spíše občasných jednotlivých spojení, navíc většinou adresovaných na jeden konktrétní SMTP server, který pro danou síť zajišťuje další doručení e-mailu.

Tento tzv. relay server umožňuje od uživatelského poštovního programu přijmout zprávu, jejíž adresát na tomto poštovním serveru nemá svoji schránku, a tuto pak dále odešle protokolem SMTP jejímu adresátovi. Takovýto server je obvykle součástí počítačové sítě firmy nebo jiné instituce, kde zprostředkovává mailovou komunikaci uživatelům této sítě. Může se také jednat o službu, kterou nabízí poskytovatel internetových služeb svým klientům. Přístup ke službám takového serveru je pak obvykle zabezpečen omezením na IP adresy používané v sítích, pro které má předávaní pošty zabezpečovat, nebo pomocí uživatelkséhо jména a hesla, kterým je nutné se autentizovat.

Často se také využívá konfigurace, kdy je pro odesílání pošty povolen pouze jeden konkrétní SMTP server. V tom případě je detekce ještě jednodušší a každý pokus o navázání TCP spojení na port 25 kamkoli jinam než na tento určený server značí pokus o rozesílání spamu nebo chybu v konfiguraci dané stanice.

3.3 Útok na web server

Webový server patří k nejčastěji provozovaným službám v prostředí internteu, proto se také velmi často stává předmětem zájmu různých útočníků. Zdařený útok tohoto typu je velmi nepříjemný a většinou po něm následuje odpojení stroje od sítě a důkladná obnova dat spojená s hledáním děr a zadních vrátek. Nečastějším vstupem do systému se stávají různé skripty vykonávané webovým serverem. Zásadním problémem bývá neošetření jejich vstupů, díky kterému dokáže utočník skriptům místo běžných a očekávaných informací předávat příkazy, které mohou být v některých případech zpracovány interpretem, jako by do skriptu patřily. Pak lze například přistoupit k SQL databázi nebo k souborovému systému serveru [3].

Bohužel takovýto útok lze jen velmi težko detekovat na úrovni síťové nebo transportní vrstvy. Požadavek ve formě URI na konkrétní zdroj dostupný na webovém servu, včetně předávání parametrů sloužících jako vstup vykonávaných skriptů, je záležitostí aplikační vrstvy. Ani statistická analýza provozu nepovede k použitelným výsledkům, protože charakteristika datového provozu se bude odvíjet od aplikace provozované na webovém serveru - jinak bude vypadat pro server poskytující víceméně statické webové stránky a jinak pro server, na kterém běží například moderní webové aplikace nebo server poskytující ke stažení objemné soubory. Navíc všechny tyto druhy provozu se často vyskytují na jednom serveru současně.

3.4 Slovníkový útok SSH

Protokol SSH (secure shell) zajišťuje šifrovaný přístup k shellu operačního systému přes počítačovou síť. Serverová část SSH běží v systému, ke kterému má být umožněno bezpečné vzálené přihlášení a na TCP portu 22 čeká na spojení od SSH klientů. Slovníkový útok spočívá v opakovaných pokusech o přihlášení k serveru SSH, kde se jako hesla posílají slova ze slovníku. Obdobou tohoto útoku je útkok hrubou silou, kde se jako heslo zkoušejí všechny možné kombinace znaků. Obě varianty útoku předpokládají, že uživatelé systému používají slabá hesla, což je bohužel v mnoha případech pravda. Ale i v případě použití kvalitních hesel, kdy tento druh útoku není nebezpečný, může být nepříjemný v tom, že zbytečně zvyšuje zatížení serveru a dochází k plnění logů záznamy o neúspěšných pokusech o přihlášení.

Tyto útoky jsou dnes nejčastěji vedeny ze sítí botnet, což jsou sítě propojených počítačů napadených nějakým druhem zákeřného softwaru (malware). Tyto počítače vyhledávají v internetu systémy s přístupným SSH serverem, který se poté snaží napadnout. Vzhledem k tomu, že po několika neúspěšných pokusech o přihlášení server většinou přeruší navázané spojení a útočník tak musí navázat nové, je možné tento druh útoku detekovat jako stále se opakující ktrátké TCP spojení z jedné IP adresy.

3.5 Kontrola dodržování bezpečnostních politik

V této kategorii jde o některé druhy síťového provozu, které sice samy o sobě nejsou považovány za bezpečnostní hrozby, můžou být ale z různých důvodů provozovatelem sítě v rámci bezpečnostní politiky zakázány. V tom případě je vhodné mít nějakou možnost kontroly jejího dodržování. Běžným příkladem může být skryté připojování neschválených zařízení do sítě s využitím technik NAT nebo využívání určitých typů síťových služeb - sdílení souborů v PtP sítích, Instant messaging, Skype atd.

Detekce v těchto případech je většinou možná. Například měnící se hodnota TTL v hlavičce IP paketu může ukazovat na použití NATu, používání různých služeb lze zase detekovat pomocí charakteristických TCP/UDP portů nebo známých cílových IP adres. Ovšem tato detekce bude značně ztížena, pokud jí „útočník“ předpokládá a bude se jí aktivně bránit (úprava TTL na jednotnou hodnotu, používání tunelů atd.).

3.6 Skenování IP adres a portů

Skenování IP adres a portů jsou techniky používané ke vzálenému zmapování sítě. Tyto techniky jsou často využívány útočníky, kteří se snaží o neautorizované prohledání sítě, s cílem získat užitečné informace k provedení dalších útoků. Takto si mohou najít využitelnou slabinu (například neaktualizovaný software nebo zneužitelné služby) na vzdáleném síťovém zařízení.

Skenování IP adres je proces, kterým jsou zjišťovány aktivní IP adresy v síti. K této činnosti se nejčastěji používá zasílání zpráv Echo Request a Echo Reply definovaných protokolem ICMP (známé také jako ping).

Skenování portů je proces, kdy se zjišťuje, které služby poskytuje dané síťové zařízení. Realizovat ho lze různými metodami v závislosti na tom, zda chceme skenovat porty protokolu TCP nebo UDP. V případě TCP protokolu se testování portů nejčastěji provádí zasáním pakteu s nastaveným příznakem SYN a následně se testují příznaky paketu vráce-

ného serverem jako odpověď (pokud server odpověděl), pro zjištění stavu portu. U protokolu UDP je odeslán běžný paket a očekává se, zda server na tomto portu odpoví, nebo vrátí pomocí protokolu ICMP ohlášení chyby.

Stejně jako v případě rozesílání spamu nebo slovníkových útoků na SSH, je tato činnost často prováděna z počítačů napadených malwarem, který toto skenování provádí pro potřeby jeho tvůrce. Z toho důvodu je možnost detekce takového chování u stanic ve sledované síti užitečná pro upozornění na tyto problematické, zákeřným softwarem napadené počítače.

Kapitola 4

Detekovatelnost hrozeb s využitím NetFow statistik

Následující část se zabývá studií bezpečnostních hrozeb z hlediska charakteristik, kterými se projevují v NetFlow statistikách. Na základě těchto zjištěných charakteristických vlastností jsou následně navrženy postupy testů a jejich parametry, které umožňují tyto hrozby v NetFlow datech automaticky detekovat.

Jak již bylo podrobně rozebráno v části věnované technologii NetFlow, zpracovávaná data mají charakter statistických informací o datových tocích v síti. Proto u většiny testů není bez znalosti konkrétních dat, která jsou v těchto tocích přenášena, detekce hrozeb zcela exaktní. Navíc u některých testů by ani znalost obsahu přenášených informací nemohla pomoci ke zcela přesnému rozhodnutí. Popisované testy jsou proto navrhovány tak, aby tvořili vhodný kompromis mezi maximální detekcí všech nebezpečných jevů v síťovém provozu a minimálním množstvím falešně pozitivních hlášení o nalezení bezpečnostní hrozby, jejichž původem byl ovšem legitimní síťový provoz, který svými vlastnostmi v NetFlow statistikách podobá detekovaným hrozbám.

4.1 Detekce počítačů rozesílajících Spam

První zpracovaný test se věnujě detekci síťových stanic rozesílajících nevyžádanou poštu – tzv. spam. Jak již bylo popsáno v předchozí kapitole, výskyt takovýchto stanic napadených škodlivým softwarem v síti, může mít pro bezpečnost a spolehlivost sítě nepříjemné důsledky.

4.1.1 Rozbor netflow

Pro stanovení charakteristických znaků šíření spamu v NetFlow statistikách byly analyzovány NetFlow data několika napadených počítačů, které rozesílaly nevyžádanou poštu prostřednictvím SMTP protokolu.

Základní společnou vlastností těchto záznamů bylo především velké možství navazovaných spojení v krátkém čase na různé cíle. Jako příklad budiž uveden jeden 8 minut trvající záznam síťové komunikace vedené z takto napadeného počítače. V těchto datech byly identifikovány všechny odchozí SMTP spojení, tedy všechny datové toky na transportním protokolu TCP směrované na cílový port číslo 25. Takto bylo zjištěno celkem 3102 odchozích SMTP spojení a to na 1831 různých cílových IP adres.

Informace o počtu navazovaných spojení z těchto NetFlow záznamů jednotlivých spamujících počítačů byly přepočítány na jednotku času a jsou shrnuty v následující tabulce:

	minimalní	průměr	maximální
Celkem spojení / minuta	62	137	387
Různých cílu / minuta	24	67	228

Počet vytvářených spojení spamujícími počítači

Dále byly zpracovány statistické informace o jednotlivých SMTP spojeních napříč těmito záznamy:

	minimalní	medián	průměr	maximální
Délka spojení [s]	0	2	5,3	247
Počet bajtů	40	431	2458	74125
Počet paketů	1	8	9	304

Vlastnosti SMTP spojení od spamujících počítačů

První tabulka jasně ukazuje vysoké množství navazovaných spojení a také vysoké množství různých cílů. I v nejmírnějším zaznamenaném případě bylo v průměru zaznamenáno 62 vytvořených SMTP spojení na 24 různých cílových adres za minutu. Poměrně vysoký rozptyl mezi zjištěnými počty navazovaných spojení může být dán jednak rozdílnou implementací spamovacího robota, případně i různou dostupnou kapacitou internetového připojení stanic v síti. Ta byla ve všech zaznamenaných případech omezena na hodnoty v rozsahu zhruba 0,5 - 2 megabity za sekundu.

Této vlastnosti je s úspěchem možné využít k detekci spamujících počítaču, protože i nejmenší zaznamenaná frekvence nově vytvářených spojení vysoce převyšuje obvyklé hodnoty, kterých lze docílit při běžné e-mailové komunikaci v domácím nebo kancelářském prostředí.

Naopak statistické informace o vytvořencýh SMTP spojeních uvedené ve druhé tabulce žádné nové charakteristiky využitelné pro požadovanou detekci spamu nepřináši. Všechny zde uvedené hodnoty budou pravděpodobně velmi podobné i u datových toků vytvářených legitimní činností.

4.1.2 Rozbor vzorových NetFlow dat

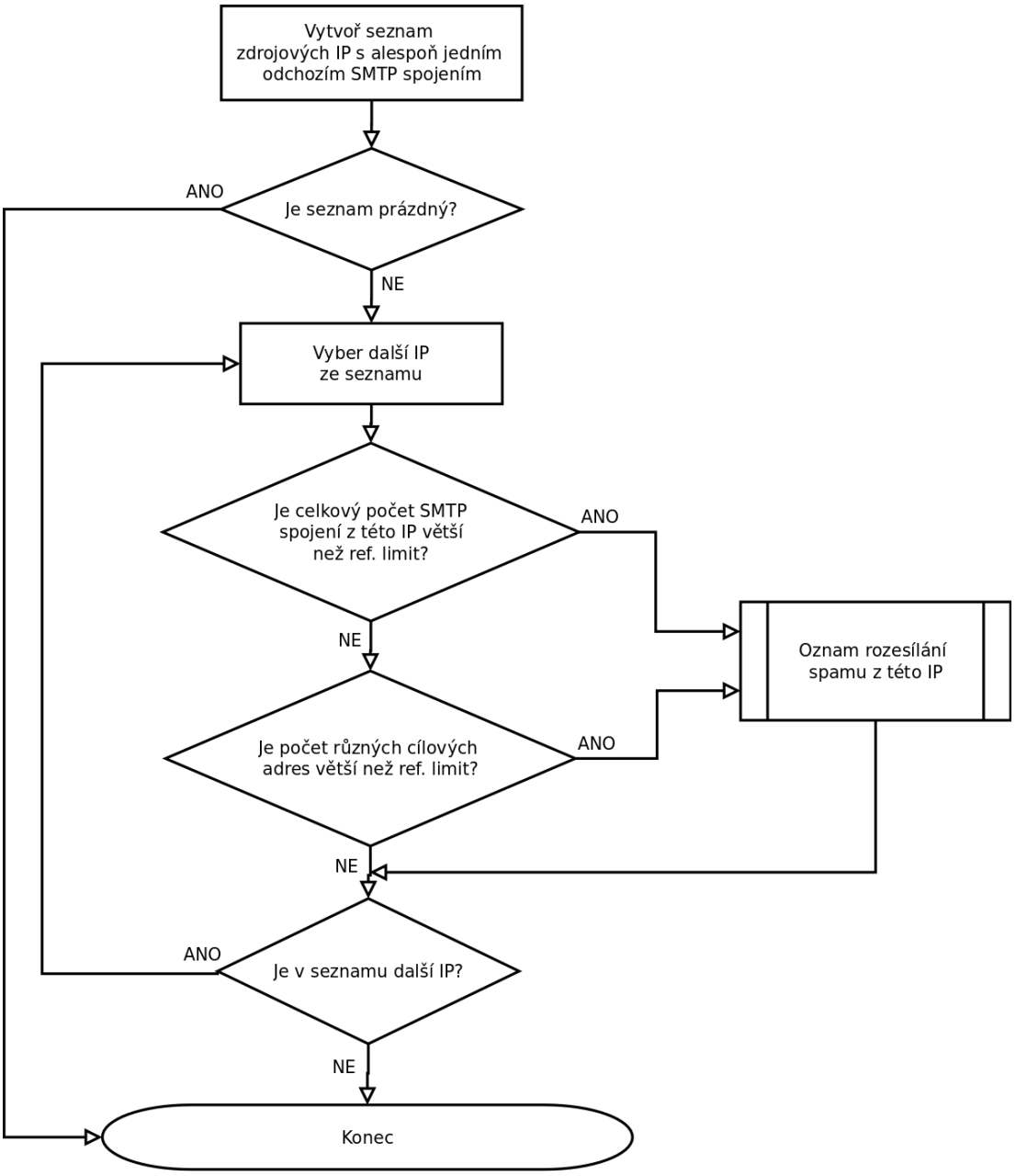
Navržený test nejdříve prohledá všechny spojení v NetFlow statistikách a vytvoří seznam zdrojových adres, ze kterých bylo uskutečněno alespoň jedno odchozí SMTP spojení - tedy takové spojení, které bylo vytvořeno pomocí protokolu transportní vrstvy TCP (číslo tohoto protokolu v příslušné položce NetFlow záznamu je 6) a byl veden na port číslo 25 (toto číslo portu je přiděleno organizací IANA pro SMTP protokol) na cílové straně. Pro každou položku v takto vytvořeném seznamu je následně zjišťován celkový počet vytvořených SMTP spojení a počet různých cílů, ke kterým byly tyto datové toky směrovány. Tyto dvě zjištěné hodnoty jsou poté srovnány s hodnotami referenčními, které byly stanoveny na základě dříve popsané analýzy chování spamovacích robotů.

Tyto referenční hodnoty byly stanoveny na 20 spojení za minutu celkem a 5 spojení za minutu na různé cíle. Pokud je některá z těchto hodnot (nebo obě současně) překročena, je právě testovaná zrojová IP vyhodnocena jako stanice rozesílající spam. Pozn.: implementace tohoto detektoru pracuje s časovým oknem pro detekci o velikosti 5 minut a proto také používá pětinásobek zde uvedených referenčních hodnot.

Navržené referenční hodnoty je samozřejmě možné podle potřeby přizpůsobit konkrétním podmínkám. Lze si například představit, že v sítí kde jsou kladeny vysoké požadavky na bezpečnost budou tyto limity nastaveny ještě přísněji a to i za cenu možné občasné falešně pozitivní detekce tohoto bezpečnostního incidentu, s následnou „ruční“ kontrolou takto označené stanice administrátorem sítě.

Opačným příkladem pak může být nasazení detektoru v síti poskytovatele internetových služeb, kde jsou výsledky využity k automatizovanému upozornění klientů na rozesílaní spamu z jejich počítače. V takovém případě je vhodné omezit pravděpodobnost falešné detekce na minimum, a výše uvedené limity nastavit benevolentněji.

4.1.3 Rozhodovací algoritmus



Rozhodovací algoritmus detekce spamujících počítačů

4.2 Detekce slovníkových útoků na SSH

Následující část se zabývá možností detekce slovníkového útoku nebo útoku hrubou silou na servery SSH. Jedná se tedy o útoky spočívající v pokusech o přihlášení, kde se jako hesla posílají slova ze slovníku nebo kde se jako heslo zkoušejí všechny možné kombinace znaků.

4.2.1 Rozbor vzorových NetFlow dat

NetFlow záznamy použité k analýze tohoto druhu útoku byly získány sledováním provozu několika SSH serverů přístupných z prostředí sítě internet pomocí NetFlow sondy. Pokusy o neoprávněný přístup k systému přes SSH vedené pravděpodobně z botnetů jsou velmi časté a tak nebyl problém nasbírat dostatečné možství dat během několika hodin. Navíc bylo možné k analýze využít i logovací soubory vytvořené sledovanými SSH démony a záznamy z nich porovnat se záznamy Netflow.

Ze zaznamenaných informacích o uskutečněných spojeních byly na základě času spojení a IP adresy klienta vybrány pouze ty, pro které existoval odpovídající záznam o neúspěšném pokusu o přihlášení v logovacích souborech SSH serveru, na který bylo spojení vedeno. Z další analýzy tak byly vyřazeny případy, ve kterých byla buď pouze zjišťována přítomnost SSH serveru na dané IP adrese, ale k pokusu o přihášení nedošlo, nebo spojení, u kterých došlo k úspěšné autorizaci oprávněného uživatele. Jak je vidět z následující tabulky, všchny takto získané záznamy se v řadě parametrů navzájem velmi podobají:

	minimalní	medián	průměr	maximální
Délka spojení [s]	3	4	4,8	12
Počet bajtů	1212	1264	1287	1476
Počet paketů	13	14	14,4	16

Vlastnosti spojení klasifikovaných jako slovníkový útok na SSH

Z tabulky je patrné že u všech zaznamenaných pokusů o přihášení bylo ve směru od útočníka k serveru vždy přeneseno 13 - 16 paketů. Tato vlastnost může být i sama o sobě poměrně spolehlivým rozpoznávacím znamením neúspěšného pokusu o přihlášení. Pro porovnání bylo analyzováno několik spojení, při kterých došlo k úspěšné autentizaci uživatele, v těchto případech se počet přenesených paketů ve sledovaném směru pohyboval okolo 25 v případě přihlášení pomocí ssh klíče a okolo 40 paketů v případě interaktivního přihlašování pomocí zadání uživatelského jména a hesla.

Dalším společným rysem všech zaznamenaných toků je délka trvnání spojení, v naprosté většině případů pohybující se v rozmezí asi 4 - 5 sekund. Zde už je ale rozptyl hodnot především směrem nahoru větší, v ojedinělých případech bylo zaznamenáno až 12 sekund trvající spojení. Doba trvání spojení navíc může být ovlivněna i konfigurací SSH serveru, především nastavenou prodlevou, po které server vrací negativní odpověď na chybné přihlašovací údaje a která se používá pro znesnadnění útoků na heslo hrubou silou.

V nashromážděných testovacích datech byly také zaznamenány případy, pro které neexistoval žádný záznam v logovacím souboru SSH serveru. Tato spojení byla v několika

případech uskutečněna ze stejných IP adres, ze kterých byly následně prováděny pokusy o přihlášení. Na rozdíl od nich se ale tato spojení vyznačovala menším počtem přenesených paketů, obvykle v rozmezí 1 - 3 pakety. Lze předpokládat, že útočník nejprve provedl otestování dostupnosti SSH serveru, např. metodou skenování portů, a teprve s určitým časovým odstupem začal provádět samotné pokusy o přihlášení.

Hlavním rozpoznávacím znakem tohoto typu útoků je však množství bezprostředně po sobě navazovaných spojení. Ve všech zaznamenaných případech docházelo k navázání nového spojení bezprostředně po ukončení předchozího, tedy asi každých 4 - 5 sekund, což byla, jak je uvedeno výše, průměrná doba trvání jednoho spojení. Tato charakteristická vlastnost tohoto druhu útoku je tedy využita jako hlavní rozhodovací podmínka v navrženém testu detekujícím tento útok.

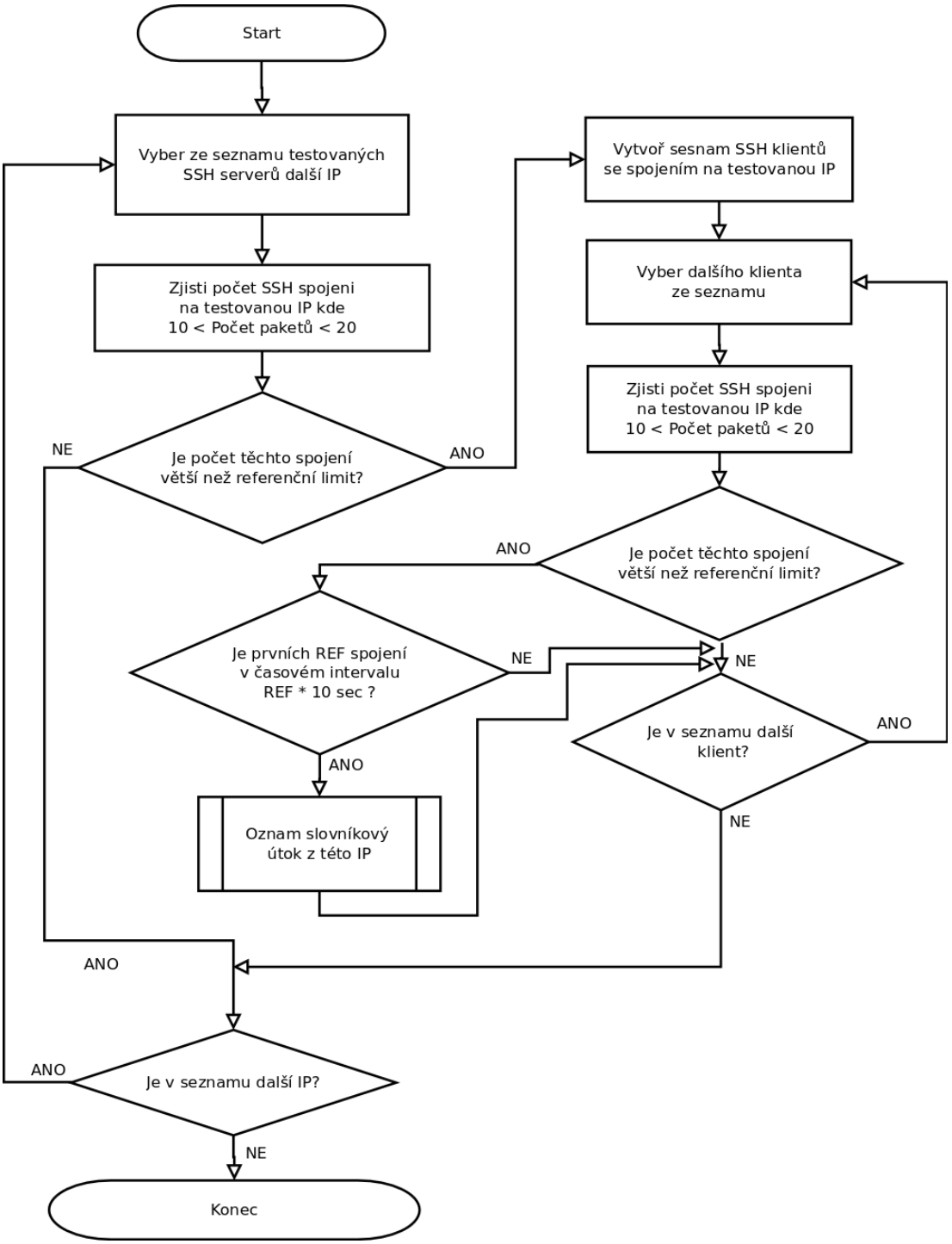
4.2.2 Návrh testu – detekce útočníků na SSH server

Navržený test předpokládá na svém vstupu seznam IP adres hostů, pro které má kontrolu na slovníkový útok provádět. Toho je v implementaci detektoru dosaženo načtením těchto adres z konfiguračního souboru testu. Pro každou z těchto IP adres se v NetFlow datech vyhledají všechny protějšky, ze kteých byla vedena na náš server SSH komunikace a sestaví se jejich seznam.

Pro každou položku z tohoto seznamu je následně zjištěn celkový počet uskutečněných spojení, která by svými parametry mohla odpovídat neúspěšnému pokusu o přihlášení. Tedy takové spojení, pro které platí, že ve směru od klienta k serveru bylo přeneseno mezi deseti až dvaceti pakety a délka spojení nepřesáhla 10 sekund. Tento počet je porovnán s referenční hodnotou. Pokud bylo uskutečněno více spojení, než kolik udává referenční hodnota, je následně proveden test rozložení těchto spojení v čase.

V chronologickém pořadí je vybráno prvních několik spojení odpovídajících výše uvedeným požadavkům, počet vybraných spojení odpovídá referenční hodnotě. Pokud zjištěný rozdíl času mezi počátkem prvního a ukončením posledního spojení z tohoto výběru menší, než referenční hodnota násobená deseti sekundami, je právě testovaná IP adresa označena jako útočník na SSH server.

4.2.3 Rozhodovací algoritmus – detekce útoku na SSH server



Rozhodovací algoritmus detekce útoku na SSH server

Jak již bylo zmíněno ve druhé kapitole mapující bezpečnostní hrozby, není při použití dostatečně silných hesel slovníkový útok nebo útkok hrubou silou nijak zvlášť nebezpečný.

Proto byla navržena ještě druhá varianta tohoto testu, která se nazaměřuje na detekci útoku na konkrétní SSH server, ale detekuje zdojové IP adresy, ze ktrých je tento druh útoku veden na libovolný cíl. Výsledky tohoto testu tak poskytují cenné informace správci sítě, protože počítač, ze kterého je tento útok prováděn, je z největší pravděpodobností napaden nějakým druhem zákeřného softwaru, podobně jako v případě počítačů rozesílajících spam.

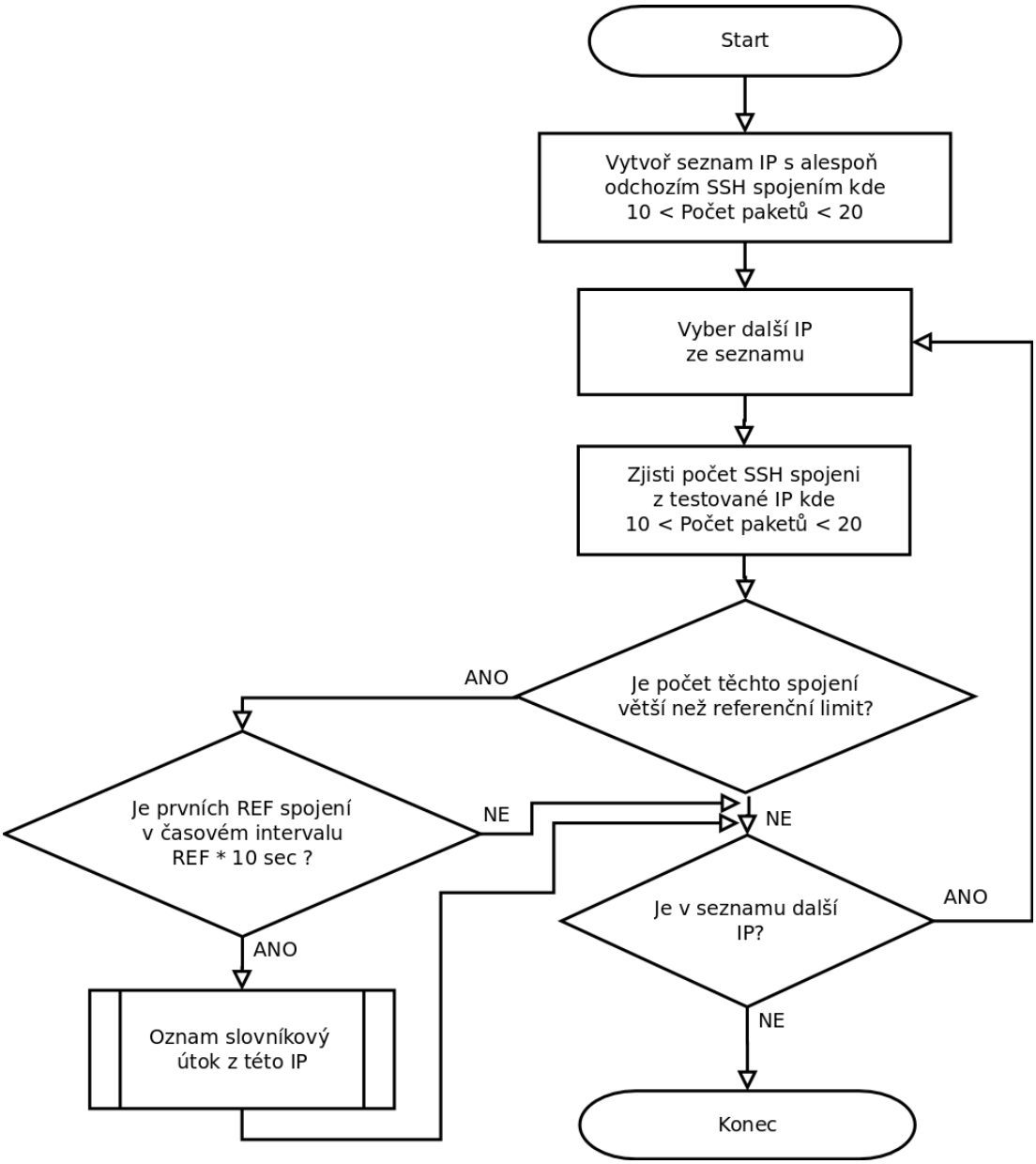
Výskyt takovýchto napadených stanic v síti, které jsou často součástí různých botnetů provádějících příkazy jejich majitele, představuje ve většině případů mnohem závažnější bezpečnostní riziko, než samotný slovníkový útok na SSH server.

4.2.4 Návrh testu – detekce útočících počítačů v síti

Detekce začíná sestavením seznamu zdrojových IP adres, ze kterých bylo uskutečněno alespoň jedno odchozí SSH spojení, tedy spojení na protokolu TCP na cílový port číslo 22. Pro každou položku z tohoto seznamu je následně zjištěn celkový počet uskutečněných spojení, které by svými parametry mohly odpovídat neúspěšnému pokusu o přihlášení. Další postup již je obdobný jako v prvním testu, detekujícím útkok na konkrétní server. V chronologickém pořadí je vybráno prvních několik takovýchto datových toků, pro které je a zjišťován rozdíl času mezi počátkem prvního a ukončením posledního spojení z tohoto výběru. Pokud všechny tyto toky proběhly během kratší doby, než je vypočítaná limitní hodnota, je právě testovaná IP adresa označena jako možný útočník.

I v tomto případě je celý postup testu je shrnut v následujicím grafickém zpracování rozhodovacího algoritmu formou vývojového diagramu:

4.2.5 Rozhodovací algoritmus – detekce útočících počítačů v síti



Rozhodovací algoritmus detekce počítačů útočících na SSH

4.3 Detekce VoIP hovoru

V této části je řešena problematika detekce VoIP hovorů realizovaných na protokolu SIP. I když VoIP hovor nelze sám o sobě považovat za bezpečnostní hrozbu, může být v některých případech možnost jeho detekce užitečná, například pro účely kontroly dodržování bezpečnostní politiky nebo pro jiné účely.

4.3.1 Rozbor vzorových NetFlow dat

Přestože byl v úvodu zmíněn internetový protokol určený pro přenos signalizace v internetové telefonii SIP (Session Initiation Protocol), není tento protokol v navržené metodě detekce VoIP hovoru nijak zahrnut. Protokol SIP vychází s osvědčeného protokolu HTTP a je mu velmi podobný. I když lze na základě čísla portu (UDP/5060 nebo TCP/5060) detekovat v NetFlow datech jeho použití, není bez znalosti obsahu přenášených paketů možné rozlišit, kdy je obsahem komunikace například přihlášení k SIP Proxy serveru, přenos konfigurace nebo vyjednávání o sestavení nebo ukončení hovoru.

Mnohem zajímavější pro možnou detekci telefonního hovoru je však protokol RTP (Realtime Transport Protocol). Tento protokol je definován v RFC 3550 a používá se pro přenos datového toku telefonního hovoru, který byl předtím vyjednaný protokolem SIP. RTP využívá jako transprtní protokol UDP a jeho obsah tvoří hlasová data zakódovaná vhodným kodekem.

Přesto že by se mohlo na první pohled zdát, že v rámci NetFlow dat nepůjde takový datový tok nijak odlišit od ostatních UDP toků, má VoIP hovor přenášený protokolem RTP má dvě základní charakteristické vlastnosti:

- Jednotlivé pakety jsou zasílány v přesných časových intervalech
- Datový tok udržuje po celou dobu trvnání konstantní bitovou rychlost

Pokud tedy máme v rámci NetFlow statistik k dispozici celkovou dobu trvání datového toku, počet přenesených paketů a bytů, a známe-li interval zasílání paketů a použitou bitovou rychlost VoIP hovorů, je možné na základě těchto informací ověřit, zda daný datový tok mohl patřit hlasovému hovoru.

Pro VoIP hovory se nejčastěji používá některý z následujících hlasových kodeků:

Název kodeku	Bitová rychlost [kb/s]	RTP výchozí ms/paket
G722	64	20
G723	5,3 nebo 6.3	30
G726-40	40	20
G726-32	32	20
G726-24	24	20
G726-16	16	20
G729	8	20
GSM	13	20
GSM-EFR	12,2	20
PCMA (G711A)	64	20
PCMU (G711U)	64	20

Vlastnosti používaných audio kodeků [4]

Jak je z tabulky patrné, RTP protokol pro všchny používané kodeky definuje výchozí metodu přenosu 20ms v jednom paketu, kromě G723, který používá pří kódování dat 30ms rámec a proto i RTP paket musí přenášet minimálně těchto 30ms. Tyto hodnoty však nejsou závazné a je možné použít i jiné hodnoty, které jsou násobky délky rámce používaného příslušným kodekem. Tento rámec má nejčastěji 10ms nebo 20ms.

Proto jsem pro účely navržení správných parametrů detektoru VoIP hovorů otestoval několik dostupných VoIP zařízení a SW klientů s cílem zjistit používanou velikost RTP paketu ve výchozím nastavení a možnosti jejího přenastavení. Zjištěné údaje shrnuje následující tabulka:

Název zařízení	výchozí ms/paket	možnost změny
AirLive VoIP-111A	20	20 nebo 30ms
Linksys SPA-2100	30	libovolně
Linksys SPA-942	20	libovolně
Well LP-302	20	nelze
Sipdroid (sw klient android)	20	nelze

Přehled možností nastavení velikosti paketu u vybraných VoIP zařízení

Na základě těchto zjištěných informací bylo rozhodnuto o nutnosti uvažovat v detektoru VoIP hovorů periodu odesílaných paketů 20 a 30ms.

4.3.2 Návrh testu

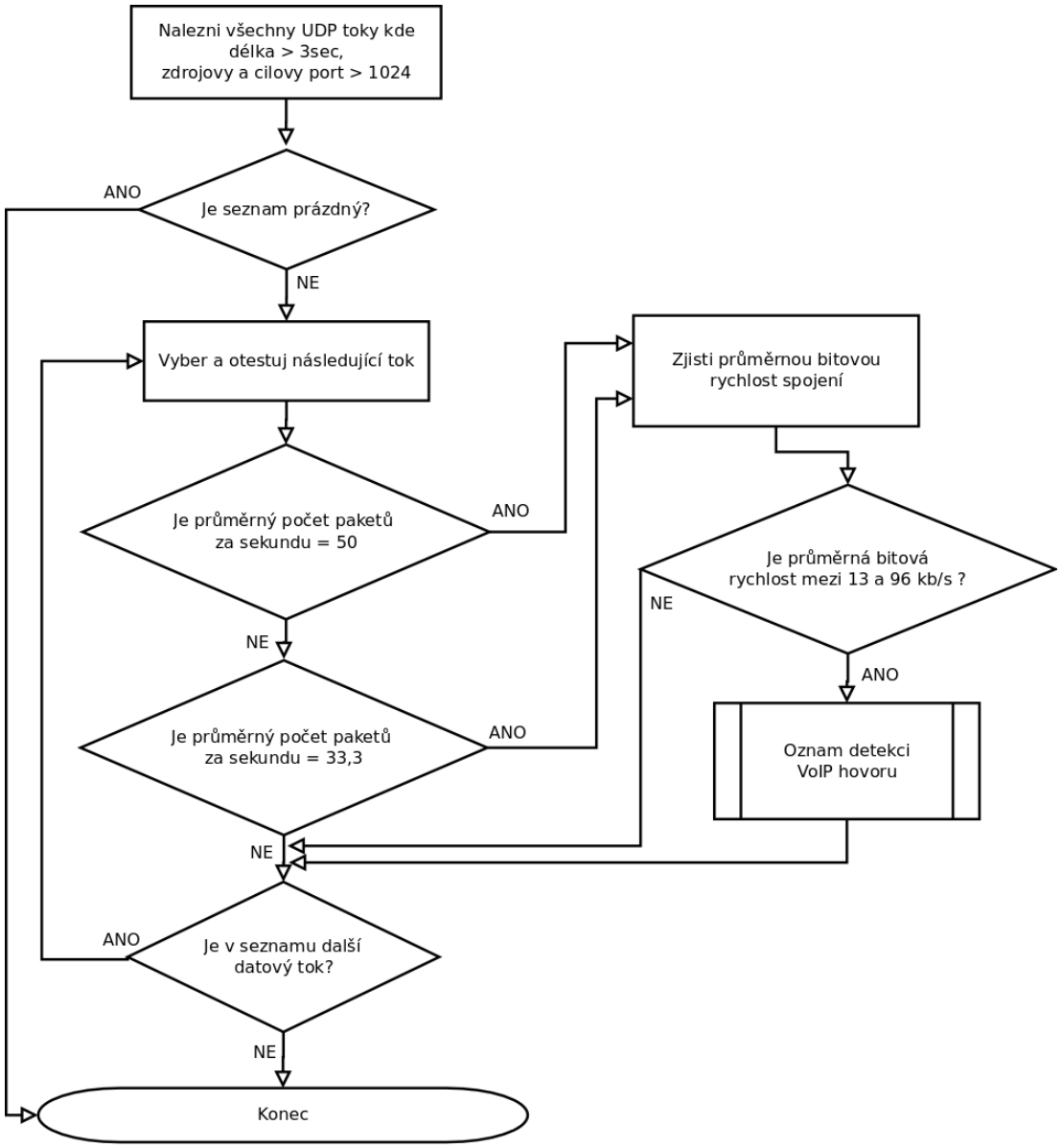
Principem navrženého testu je analýza všech UDP datových toků v NetFlow datech, při které se zjišťuje, jestli zkoumané spojení odpovídá svými vlastnostmi datovému toku generovanému VoIP provozem. Především je tedy zjišťováno, zda celkový počet přenesených paketů v průběhu toku odpovídá situaci, kdy jsou jednotlivé pakety odesílány přesně v 20-ti nebo 30-ti milisekundových intervalech. Pokud je tato podmínka splněna, je dále ověřeno jestli i celkový počet přenesených bytů není v rozporu s hodnotami bitových rychlostí v oblasti VoIP telefonie běžně používaných kodeků. Poud jsou tyto podmínky splněny, je velmi pravděpodobné, že testovaný datový tok patřil probíhajícímu telefonnímu hovoru.

Navržený test tedy nejprve hledá v NetFlow záznamech všechny datové toky na protokolu UDP, které trvaly alespoň po dobu 3 sekund. Tato minimální hranice délky detekovaného hovoru byla zavedena proto, že při kratší délce spojení začíná výrazným způsobem stoupat pravděpodobnost nalezení datových toků, které svými parametry splní výše uvedené podmínky, ale které nebyly vytvořeny VoIP provozem.

Následně je pro každý takovýto tok vypočítána průměrná hodnota odeslaných paketů za sekundu, tedy celkový počet paketů dělený hodnotou udávající délkou spojení v sekundách. Výsledek je následně porovnán s hodnotami 50 a 33,3, tedy očekývaným výsledkem pro RTP datový proud s periodou zasílaných paketu 20ms resp. 30ms. Při tomto porovnání je tolerována případná malá odchylka od očekávaných hodnot, například pro případ občasné ztráty nebo duplikace paketu při přenosu a podobně.

Pokud je předchozí podmínka pro označení datového toku jako VoIP hovoru splněna, následuje ještě ověření, zda celková přenesená data během spojení odpovídají hodnotám očekávaným pro RTP proud s bitovou rychlostí určenou použitým kodekem. Pro výpočet výsledné bitové rychlosti je nutné k hodnotám uváděným v přehledu používaných kodeků připočíst ještě velikost RTP, UDP a IPv4 nebo IPv6 hlaviček, to vše ve variantě pro 20 i 30ms paket. Pokud by měl být výsledný objem přenesených dat počítán a porovnáván pro všechny tyto varianty, bylo by toto hledání značně neefektivní. Navíc při nutném tolerování určité možné odchylky u každé z těchto variant, by ve výsledku byla akceptována víceméně jakákoliv hodnota v rozmezí bitových rychlostí uvažovaných kodeků. Z toho důvodu je prováděno pouze ověření, že počet přenesených bytů v datovém toku odpovídá bitové rychlosti v rozmezí 13 - 96 kb/s. Do tohoto rozsahu se vejdou všechny uvažované kodeky s datovým tokem 5.3 - 64 kb/s s připočítáním všech možných hlaviček ostatních protokolů.

4.3.3 Rozhodovací algoritmus



Rozhodovací algoritmus detekce VoIP hovorů

4.4 Detekce použití PtP sítě

Poslední navržený test se zabývá možností detekce použití peer-to-peer sítí pro distribuci souborů. Hlavní pozornost byla zaměřena především k protokolu BitTorrent, který je v současnosti pravděpodobně nejpoužívanějším protokolem tohoto druhu.

4.4.1 Rozbor vzorových NetFlow dat

K analýze charakteristik datových toků generovaných peer-to-peer (PtP) sítí na protokolu BitTorrent byly vytvořeny NetFlow záznamy komunikace počítače, ze kterého kromě běžícího klentského programu sítě BitTorrent nebyl v půběhu záznamu generován žádný jiný síťový provoz. Pro testy byly použity dva různé klientské programy sítě BitTorrent běžící na různých platfomách, aby se minimalizoval vliv případných implementačních specifik konkrétního programu na charakter datových toků. Polovina testovacích záznamů byla vytovřena z provouzu generovaného bittorrentovým klientem uTorrent běžícího na platformě Microsoft Windows a druhá polovina záznamů popisuje provoz vytvářený programem Deluge na platformě Linux.

Protokol BitTorrent používá ke komunikaci oba nepoužívanější transportní protokoly – TCP i UDP. Na rozdíl od protokolů jako SMTP nebo SSH, se kterými bylo pracováno v návrhu předchozích testů, není konkrétní datový tok vytovřený použitím PtP sítě v NetFlow datech možné jednoduše rozlišit. BitTorrent nepoužívá pevně dané čísla portů, spojení jsou výtvářena na náhodné porty ze skupiny registrovaných a dynamických/privátních portů – tedy čísla portů od 1024 do 65534. Při hledání význačných znaků těchto spojení, se kterými by mohl navržený detektor pracovat, proto byla věnována pozornost především velkému množství vytvářených toků, s velkým množstvím různých cílových adres, které je patrné už při prvním pohledu na obsah získaných NetFlow záznamů.

Původní BitTorrent protokol předpokládá existenci jistého centrálního prvku sítě, tzv. Trackeru. Tracker v tomto systému neposkytuje samotná data, tedy části souborů distribuovaných touto sítí, ale místo nich udržuje seznam IP adres klentů, kteří vlastní jednotlivé kousky souborů a také zprostředkovává spojení mezi zúčastněnými počítači a řídí provoz.

V poslední době se ale stále častěji používá i rozšíření protokou BitTorrent o funkci DHT, díky kterému může být tato síť zcela decentralizovaná. DHT (Distributed Hash Table) slouží k hledání a výměně protějšků, kteří mají k dispozici požadovaná data bez toho, aby byl pro tuto činnost použit centrální prvek sítě (Tracker) jako v případě, kdy je tato funkce vypnuta. DHT je založena na principu hashovací tabulky, která udržuje stejné inforamce o dostupnosti částí šířeného souboru jako tracker v klasické variantě. Tato tabulka je distribuována mezi všemi klienty sdílejícími stejný datový soubor.

Bylo předpokládáno, že charakteristika zaznamenaného síťového provozu z testovaných klientů sítě Bittorrent může být do značné míry ovlivněna použitím nebo nepoužitím funkce DHT. Proto byly všechny záznamy komunikace pořízeny jak ve variantách se zapnutoým i vypnutým rozšířením protokolu DHT. Rozvněž oděleně byly zaznamenávány a porovnávány dva režimy činnosti – prvním z nich je stav kdy nebyl aktivně stahován žádný soubor, ale pouze několik již dříve stažených souborů bylo dále prostřednictvím této sítě distribuováno ostatním klientům. V terminologii protokolu BitTorrent je tento stav nazýván jako „seed“ souboru. Ve druhém případě byl naopak pouze jeden soubor stahován a s vyjímkou dosud stažených částí tohoto souboru nebyl ostatním klientům poskytován žádný jiný obsah.

Zjištěné průměrné počty nově vytvářených spojení za minutu shrunuje následující ta-

bulka:

	TCP / min	UDP / min	Celkem / min
Seed souborů s DHT	132	482	614
Seed souborů bez DHT	113	313	426
Stahování souboru s DHT	185	205	390
Stahování souboru bez DHT	48	25	73

Počet vytvářených spojení v PtP komunikaci

Obdobná tabulka shrnující počet různých cílu, se kterými byla vedena komuikace:

	TCP / min	UDP / min	Celkem / min
Seed souborů s DHT	45	347	392
Seed souborů bez DHT	97	180	227
Stahování souboru s DHT	34	91	125
Stahování souboru bez DHT	9	8	17

Počet různých cílových IP adres v PtP komunikaci

Jak bylo předpokládáno, povolení funkce DHT v protokolu má zjistitelný vliv na frekvenci, s jakou jsou vytvářeny nové spojení i na počet cílů komunikace. Ve všech případech došlo po jejím zapnutí k nárustu hodnot sledovaných údajů, což je dáno zvýšenou režií protokolu, nutnou pro správu distribuované hashovací tabulky, oproti případu, kdy je místo ní použita komunikace s centrálním prvkem, který tyto informace spravuje.

Pro úplnost je uvedena ještě tabulka shrnující tyto datové toky s ohledem na počet přenesených paketů:

	minimalní	medián	průměr	maximální
TCP toky	1	3	51,1	36791
UDP toky	1	1	18,1	123836

Statistika počtu paketů v přenášených tocích

Přestože tedy není bez znalosti obsahu přenášených paketů možné s jistotou určit příslušnost konkrétního spojení k provozu PtP sítě, je možné detekovat použití těchto síti na konkrétním počítači podle množství vytvořených spojení na velký počet různých cílů. V případě, kdy žádná PtP síť není využívána, není tento charakteristický jev v provozu generovaném počítači v běžném domácím nebo kancelářském prostředí obvyklý. Nejčastější typy spojení v těchto podmínkách, tedy především HTTP a HTTPs protokol a nebo třeba DNS, FTP, POP/IMAP a SMTP, lze již předem vyloučit omezením výběru na čísla portů větší než 1024 a u ostatních služeb většinou nedochází ke komunikaci s velkým množstvím různých cílových adres v krátkém čase.

4.4.2 Návrh testu

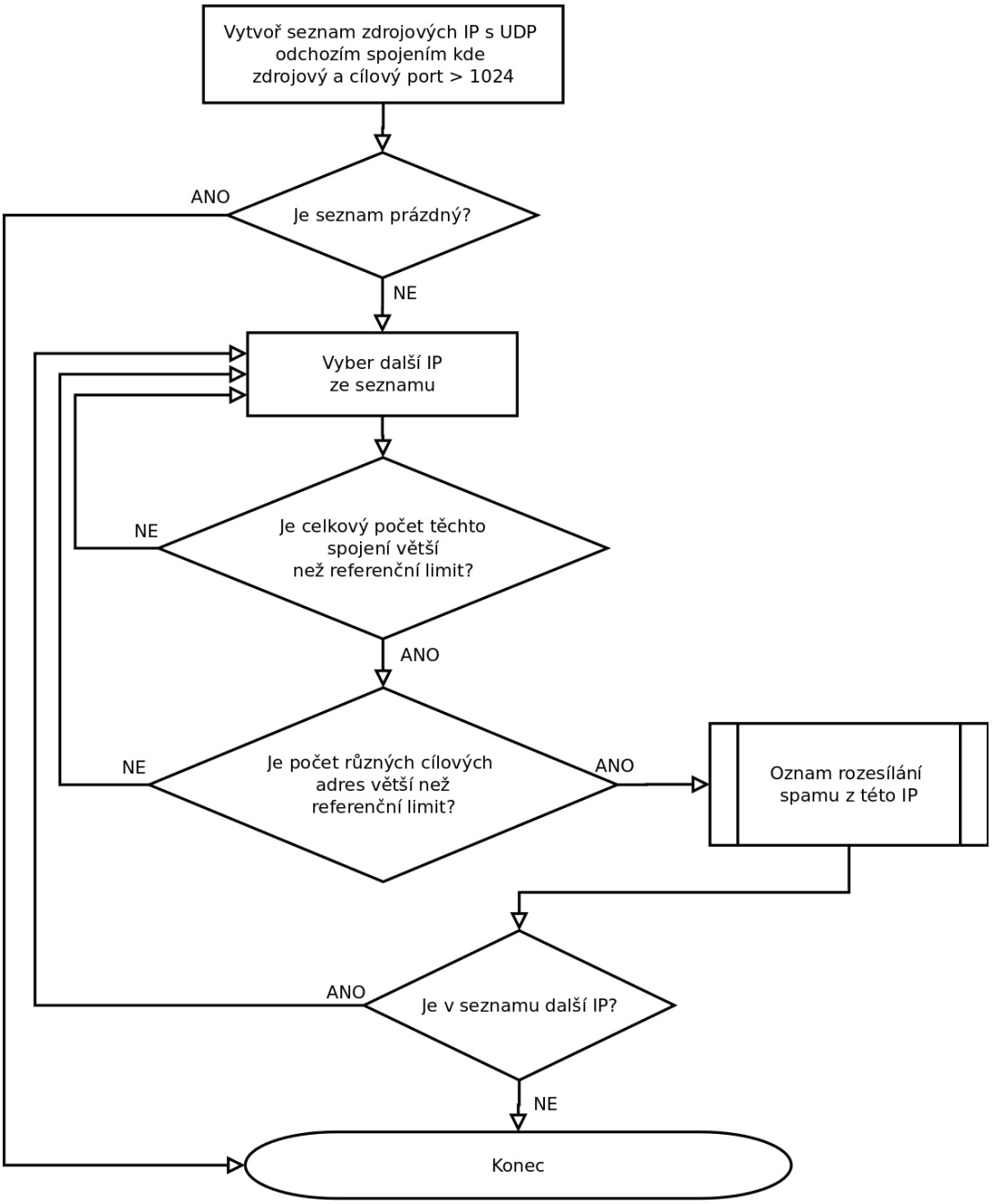
Navržený test nejdříve prohledá všechny spojení v NetFlow statistikách a vytvoří seznam zdrojových adres, ze kterých byl ve zpracovávaném časovém rámci počet uskutečněných spojení větší než stanovený referenční limit. Hledány jsou pouze takové toky, které byly uskutečněny na protokolu TCP nebo UDP a kde cílový i zdrojový port je větší než 1024. V případě, kdy předpokládáme, že by mohla z nějakého důvodu existovat snaha uživatelů vyhnout se prováděné kontrole na detekci PtP sítí, lze u zdorjového portu podmínku čísla většího než 1024 vypustit. To znemožní obejítí detekce nastavením takového portu jako preferovaného zdrojového portu v klientské aplikaci sítě BitTorrent.

Pro každou položku v takto vytvořeném seznamu je následně zjišťován počet různých cílů, se kterými byla ve sledovaném období vedena komunikace. Takto zjištěný počet cílů je následně porovnán s druhou referenční hodnotou, která udává maximální povolený limit této veličiny.

Obě předchozí referenční hodnoty byly stanoveny na základě dříve popsané analýzy provozu peer-to-peer sítě BitTorrent. Pro celkový počet navázaných spojení je zde počítáno s hodnotou 350 a pro počet různých cílů je hodnota stanovena na 75. Obě pro časový rámec 5 minut, se kterým je v tomto testu uvažováno jako s výchozím. Navržené referenční hodnoty je samozřejmě možné podle potřeby přizpůsobit konkrétním podmínkám.

Celý postup testu je shrnut v následujicím grafickém zpracování rozhodovacího algoritmu formou vývojového diagramu:

4.4.3 Rozhodovací algoritmus



Rozhodovací algoritmus detekce použití peer-to-peer sítí

4.5 Společné vlastnosti navržených testů

Všchny tyto testy bezpečnostních hrozeb byly navrženy tak, aby k jejich správné činnosti vždy stačila NetFlow data zaznamenávající datové toky pouze v jednom směru. To je výhodné zejména v případě, kdy tato data exportujeme ze směrovače nebo jiného zařízení s omezeným výpočetním výkonem. Další nezanedbatelnou výhodou tohoto řešní je značné snížení velikosti dat, které je nutné uchovávat a zpracovávat.

Pro všechny implementované detektory, s vyjímkou detekce slovníkového útoku na zadaný SSH server, tedy postačí data zaznamenaná v odchozím směru od počítačů pro které má být prováděna detekce popisovaných případů. V případě detektoru pro slovníkový útok (v první popisované variantě), postačí rovněž jednosměrná data, ale v opačném směru – sledovaný SSH server by měl být v těchto tocích veden jako cílová IP adresa.

Všechny detektory samozřejmě pracují i na NetFlow datech s informacemi o datových tocích v obou směrech. V takovém případě je ale vhodné v konfiguraci jednotlivých testů explicitně určit zdorjové IP adresy pro které má být detekce prováděna, jinak se budou testy provádět pro všechny zdrojové IP adresy v těchto datech, což ve většině případů nebude vhodné.

Kapitola 5

Implementace

V této kapitole je blíže popsána vytvořená prototypová aplikace implementující testy pro detekci bezpečnostních hrozeb, které byly navržený v předchozí kapitole. V první části jsou blíže představeny dva softwarové produkty, kterých výsledná aplikace ke své činnosti využívá. Jde o implementaci NetFlow kolektoru Flowd a relační databázový systém MySQL. Aplikaci detekující bezpečnostní hrozby z NetFlow dat, která byla vytvořena jako součást řešení této diplomové práce, se pak věnuje druhá část této kapitoly.

5.1 Flowd

Pro získávaní dat je využita implementace NetFlow kolektoru Flowd. Tento software je šířen pod obdobou BSD licence, jeho autorem je Damien Miller a domovská stránka projektu je http://www.mindrot.org/projects/flowd/ . Flowd je určen pro platformy Linux a OpenBSD, autor však nevylučuje funkci i na jiných unixových systémech, na kterých ale nebyl testován. Podporován je NetFlow protokol ve verzích v.1, v.5, v.7 a v.9 a spolupracuje tak se všemi NetFlow exportéry, které používají některou z těchto verzí protokolu. Plně podporován je rovněž síťový protokol IPv6 a to jak zpracování informací o IPv6 datových tocích, které umožnuje NetFlow od verze 9, tak i jako síťový protokol pro transport dat z exportéru na kolektor.

Flowd rovněž disponuje filtrem příchozích NetFlow datagramů, který používá obdobnou syntaxi pravidel jako paketový filtr PF v OpenBSD. Takto lze už před uložením do souboru filtrovat informace o tocích na základě zdrojových a cílových IP a portů. Rovněž lze řešit povolování a zakazování příjmu dat z různých exportérů podle jejich IP adresy, ale třeba i na základě různých časových intervalů. Informace o datových tocích jsou ukládány do souboru v úsporném binárním formátu a navíc je v konfiguraci kolektoru možné vybrat, které položky NetFlow záznamu se budou ukládat a které ne. Tím je možné dále snížit velikost vytvářených souborů.

Flowd umožňuje běh na pozadí jako unixový démon. Nastavení je řízeno konfiguračním soborem, který je zpracován při startu programu, za běhu je možné ovlivnit jeho chování zasíláním signálů. Tímto způsobem je možné například vynutit znovunačtení konfiguračního souboru, uzavření původního datového souboru a zahájení zápisu NetFlow dat do nového souboru, vypsání statistických informací o běhu programu nebo jeho ukončení.

Kromě ukládání přijatých dat do souboru je možné i jejich předávání přes místní (unixový) socket. Tím je umožněno například okamžité zpravání přijatých dat nějakou aplikací provádějící jejich analýzu.

Flowd také poskytuje programátorské rozhraní pro scriptovací jazyky Perl a Python, pomocí kterého je možné číst a parsovat zaznamenaná data.

Posledních dvou zmiňovaných vlastností je využito v rámci řešení této diplomové práce. Aby mohla být data získaná z NetFlow kolektoru zpracována vytvořenou aplikací pro detekci hrozeb, je využíván script napsaný v jazyce Perl, který čte data přijatá kolektorem z datového souboru nebo z unixového socketu a ukládá je do SQL databáze.

5.2 Databáze MySQL

Implementovaný detektor bezpečnostních hrozeb využívá pro uložení a práci s daty získanými z NetFlow kolektoru relační databázi MySQL.

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB, nyní vlastněný společností Sun Microsystems, dceřinnou společností Oracle Corporation. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování - je k dispozici jak pod bezplatnou licencí GPL, tak pod komerční placenou licencí.

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. SQL je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace Linux, MySQL, PHP a Apache jako základní software webového serveru („technologie LAMP“).

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, triggery, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet.[6]

Použití databázového sytému pro uložení a zpracování NetFlow dat přináší především výhodu velice rychlého vyhledávání v datech na základě nejrůznějších kritérií. Rychlosti vyhledávání, které toto řešení dosahuje by bylo velice obtížné dosáhnout implementací přímého čtení NetFlow záznamů z datového souboru ve formátu, ve kterém je ukládán kolektorem.

Dalším významným přínosem tohoto řešení je také možnost využití dotazovacího jazyka SQL, pomocí kterého je možné velmi snadno pokládat i komplikované dotazy k nalezení požadovaných dat přesně odpovídajících nejrůznějším požadavkům. Využít je možné i agregační funkce SQL, coč jsou speciální statistické funkce, které databázový systém podporuje. Ty umožňují získat souhrnné údaje, jako aritmetický průměr, součet, maximum, minimum, nebo počet hodnot.

Při návrhu aplikace i všech implementovaných testů byl proto kladen důraz na maximální možné využití těchto vlastností tak, aby co největší čás zpracování dat byla prováděna přímo databázovým systém a pouze minimum dat bylo nutné kopírovat do prostředí aplikace a zde zpracovávat. Nezastupitelnou roli hrál databázový systém a jazyk SQL také ve fázi analýzy vzorových NetFlow dat a návrhu testů. S využitím nástroje phpMyAdmin pro jednoduchou správu obsahu databáze prostřednictvím webového rozhraní bylo možné v nashromážděných NetFlow datech snadno hledat charakteristické vlastnosti jednotlivých

bezpečnostních hrozeb a testovat jednotlivé kroky navrhovaných detekčních testů ještě před jejich implementací.

5.3 Popis implementace

Navržená aplikace byla implementována v programovacím jazyce Java. Java je objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems a v současnosti je jedním z nejpoužívanějších programovacích jazyků na světě. Výhodou je, že aplikace vytvořené v tomto jazyce jsou přenositelné na libovolný operační systém a libovolnou architekturu, pro kterou existuje vyrtuální stroj Javy (JVM - Java Virtual Machine).

Jednou z nejčastěji prováděných činností implementované aplikace je komunikace s databází. K tomu účelu Java poskytuje jednotné programátorské rozhraní JDBC (Java Database Connectivity), přes které je umožněn přístup k libovolnému databázovému systému, pro který existuje JDBC ovladač. Pro databázi MySQL použitou v tomto projektu takový ovladač existuje a je možné ho pod názvem MySQL Connector/J získat na stránkách projektu MySQL.

Výsledná aplikace je distribuována v Java archivu `detektor.jar`. Tento soubor je možné spustit v systému, na kterém je k dispozici virtuální stroj Javy. Jediným parametrem je cesta ke konfiguračnímu souboru aplikace, který obsahuje především údaje nutné pro připojení k databázi a určení adresáře s konfiuračnímy soubory jednotlivých prováděných testů.

Běh programu je následně řízen konfiguračními soubory testů. Každý test je definován vlastním konfiguračním souborem, ve kterém je především uveden typ testu - tedy detekce spamujících počítaču, detekce použití PtP sítí, detekce VoIP hovorů nebo detekce slovníkového útoku na SSH. Dále zde mohou být určeny další parametry, specifické pro každý typ testu. Přitom jeden typ testu může být použit i ve více konfiguračních souborech, pokaždé s jinými parametry. To je užitečné především pro účely testování, kdy lze spustit v jednom běhu aplikace určitý test vícekrát a sledovat chovaní pro různě nastavené parametry testu. Podrobný popis konfiguračních souborů pro jednotlivé testy je uveden v příloze 1 této práce.

Spuštěná aplikace provede všechny zadané testy, zjištěné výsledky vypíše do souboru nebo na standardní výstup (lze určit v konfiguraci), a poté se ukončí. U aplikace se předpokládá pravidelné spouštění v určitém časovém intervalu, např. každých 5 minut. Toho lze v systémech na bázi Unixu dosáhnout například konfigurací plánovače úloh Cron, který zajišťuje automatizované spouštění příkazů v určitý čas. V takovém případě jsou testy prováděny pouze na datech, která byla do databáze přidána od posednícho spuštění aplikace. Rovněž je možné nastavit čas, po kterém budou starší data z databáze mazána.

Toto chování lze v konfiguraci testů zakázat a provátět testy na všech datech obsažených v databázi. V tom případě je ale nutné počítat s tím, že testy, které byly navrženy pro práci na datech za určitý časový rámcec - konkrétně hledání spamovacích robotů a detekce PtP sítí, nemusí dávat správné výsledky. Naopak testy detekující VoIP hovory a Slovníkový útok mohou správně fungovat na datech za libovolně dlouhé období.

5.3.1 Schéma procesu zpracování NetFlow dat

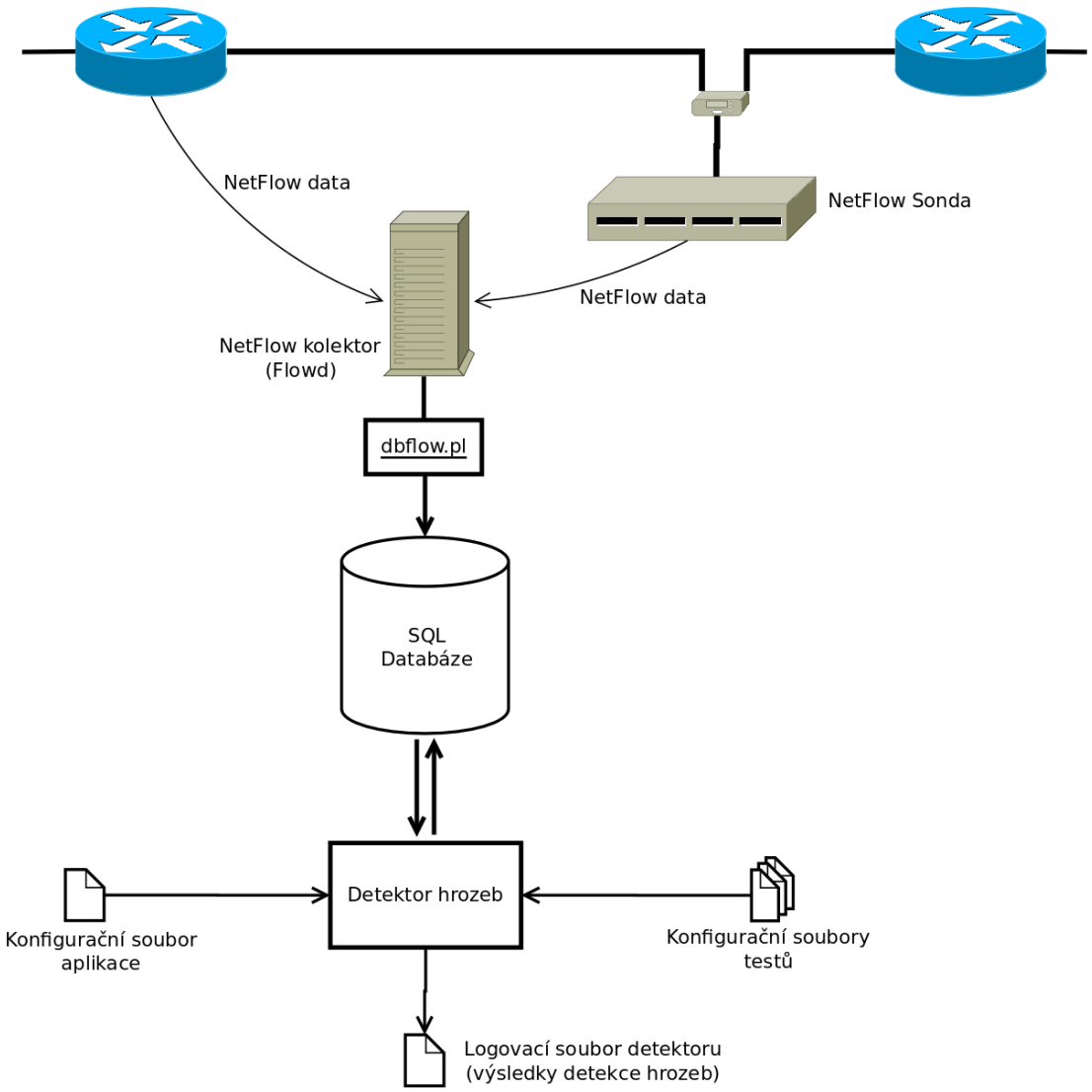


Schéma celého procesu zpracování NetFlow dat

Kapitola 6

Dosažené výsledky a možnosti rozšíření

6.1 Testovací NetFlow data

Všechny navržené testy bezpečnostních hrozeb byly primárně testovány na stejných NetFlow datech, které byly využity pro účely analýzy charakteristických vlastností těchto hrozeb. Tyto vzorky dat byly získány buď záznamem nalezeného reálně se vyskytujícího případu v síti (rozesílání spamu, slovníkový útok), nebo byly pro potřeby získání těchto vzorků přímo vytvořeny (záznam VoIP hovoru s různými parametry, použití peer-to-peer sítě).

Následně byla implementovaná aplikace testována na NetFlow záznamech pocházejících z reálného provozu počítačové sítě občanského sdružení PernštejnNET. Tato síť slouží především k zajištení připojení do sítě Internet členům sdružení, kterých je v současnosti přibližně 200. Testovací NetFlow data byla exportována z routeru, přes který prochází téměř celý provoz této sítě, s vyjímkou několika serverů. Zaznamenáno bylo asi 7 hodin provozu v době relativně silného provozu v síti, přižemž byly pořizovány záznamy o datových tocích směřujících z této sítě do sítě Internet. Opačný směr datových toků zaznamenáván nebyl, jelikož pro správnou funkci navržených testů není nutný. Zakladní vlastnosti pořízeného testovacího záznamu shrnuje následující tabulka:

Délka záznamu	7h 12m
Počet zaznamenaných datových toků	711 889
Celkový počet přenesených dat	16,3 GB
Celkový počet přenesených paketů	29,2 mil
Počet komunikujících zrojových IP adres	209
Počet unikátních cílů v síti Internet	117 942
Velikost získaných NetFlow dat	80,6 MB

Vlastnosti testovacího NetFlow záznamu komunikace

6.2 Výsledky implementovaných detektorů

Následuje popis testování jednotlivých implementovaných detektorů na dostupných testovacích datech a jejich výsledky v detekci bezpečnostních hrozeb.

6.2.1 Detekce spamu

Implementovaný detektor spamujících počítačů při testech dokázal správně označit všechny případy této činnosti zaznamenané v testovacích NetFlow datech. Při testu na reálných datech z provozu počítačové sítě žádný takovýto případ zaznamenán nebyl, přestože se v těchto datech celkově vyskytovalo několik desítek SMTP spojení. Tento výsledek byl porovnáván s výsledkem systému pro detekci spamujících stanic používaným v síti PernštejnNET, který pracuje na jiném principu, než detekce z NetFlow dat. Ani tento systém nezaznamenal v průběhu doby, kdy byl testovací záznam NetFlow dat pořizován, žádný spamující počítač v síti. Výsledeky detektoru lze tedy ve všech případech považovat za správné.

6.2.2 Slovníkový útok na SSH

Tento typ bezpečnostní hrozby jako jediný nebyl testován na vzroku dat z běžného provozu sítě PernštejnNET. Implementovaná varianta tohoto detektoru, která provádí detekci útoku na konkrétní SSH server jednak předpokládá NetFlow data z opačného směru, než ve kterém byly pořízeny testovancí data, a navíc se ve sledované části sítě nenacházel žádný SSH server, který by byl přístupný ze sítě Internet.

Proto byl za účelem tohoto testu zřízen testovací stroj s běžícím SSH serverem, který byl pro zvýšení četnosti pokusů o útok z internetu dostupný na sedmi různých IP adresách. Vešterá komunikace tohoto stroje byla zaznamenávána NetFlow sondou. V rámci možností pak bylo provedeno i několik přihlášení na tento server s použitím správného uživatelského jména a hesla nebo ssh klíče. Výsledky detektoru na těchto datech byly porovnány s dostupnými logovacími soubory SSH serveru. V průběhu testu bylo zaznamenáno několik sérií pokusů o přihlášení z internetu, všechny z nich byly detektorem správně ohlášeny.

6.2.3 Detekce VoIP hovorů

Detektor VoIP hovorů dokázal správně označit několik testovacích hovorů zaznamenaných v NetFlow datech, které byly zaznamenány pro určení charakteristik takovýchto datových toků. V reálných datech z provozu celé počítačové sítě pak našel všechny hovory, které byly v průběhu záznamu záměrně vytvořeny z IP telefonu se známou IP adresou. Vedle toho označil i několik dalších spojení, které byly vedeny z IP adres vyhrazených v síti PernštejnNET pro provoz VoIP telefonů a je tedy velice pravděpodobné, že tyto toky patřily opravdu VoIP hovoru. Stejně tak lze za VoIP hovor považovat jedno zaznamenané spojení, které bylo vedeno z IP, která není vyhrazena pro IP telfony, ale tento tok směroval na IP adresu patřící známemu českemu poskytovateli VoIP služeb. Dále byly označeny asi 3 toky, u kterých nebylo možné potvrdit ani vyvrátit, zda se původně jednalo o VoIP hovor.

6.2.4 Detekce použití PtP sítí

Stejně jako předchozí testy, i detektor použití peer-to-peer sítě pracoval správně na datech, podle kterých byl prováděn jeho návrh. V NetFlow datech z provzu celé sítě pak označil několik zdrojových IP adres, u většiny z nich ale nebylo možné ověřit, zda k použití PtP sítě

skutečně došlo. Správně ale byly v těchto datech označeny oba počítače, na kterých byl po dobu záznamu záměrně provozován klient sítě BitTorrent přenášející data. V průběhu testu také nebyla označena žádná IP adresa z těch, o kterých bylo bezpečně známo, že neslouží k provozu PtP sítě. Takových adres je ale z celkového počtu pouze malý zlomek.

Souhrnem lze tedy říci, že všechny testované detektory ve všech případech, které bylo možné nějakým způsobem nezávisle ověřit, rozhodovaly správně. V průběhu testů se ale vyskytlo i několik případů, které nebylo možné ověřit a správnost rozhodnutí detektoru o nalezení příslušné bezpečnostní hrozby je tak nejistá.

6.3 Možná rozšíření projektu

Jako první možnost rozšíření se určitě nabízí doplnění detektorů pro další bezpečnostní hrozby. Vhodným kandidátem by mohl být například detektor počítačů provádějících skenování portů. I tato činnost, podobně jako rozesílání spamu nebo slovníkové útoky, je ve většině případů vedena z počítačů napadených nějakou formou zákeřného softwaru. Takovýto detektor by tak vhodně doplňoval existující detektory spamu a slovníkových útoků, se kterými by tvořil, především pro správce sítí různých institucí a firem, zajímavý funkční celek detekující problematické počítače uvnitř sledované sítě.

Další možnou oblastí rozšíření je otestování a případné doplnění podpory síťového protokolu IPv6 do aplikace. Ve fázi implementace aplikace bylo sice s podporou tohoto protokolu částečně počítáno, ale vzhledem ke skutečnosti, že NetFlow exportér používaný k získání všech testovacích dat IPv6 nepodporuje, nebyla výsledná aplikace na jeho podporu nijak testována. Vzhledem k dosavadnímu velmi malému rozšíření IPv6 protokolu by ale pravděpodobně byl problém získat vhodná testovací data se vzorky bezpečnostních hrozeb, pro které byly tyto detektory řešeny.

Kapitola 7

Uživatelská příručka

Předposlední část této práce se zabývá seznámením se s demonstračím programem a způsobem jeho ovládání.

7.1 Instalace

Na přiloženém datovém nosiči se v adresáří `detektor` nachází vše potřebné pro zprovoznění vytvořeného programu:

- `flowd-0.9.1.tar.gz` - tento soubor obsahuje distribuci NetFlow kolektoru Flowd, který je v tomto projektu využíván k získávání NetFlow dat z exportérů. Aktuální verzi programu je také možné získat na adrese `http://www.mindrot.org/projects/flowd/`. V archivu je obsažen i detailni postup instalace tohoto programu. Pro otestování vytvořeného detektoru na již existujících NetFlow datech, není instalace tohoto kolektoru nutná, s vyjímkou rozhraní pro jazyk Perl, které je nutné pro čtení zaznamenaných dat.
- `dbflow.pl` - skript v jazyce Perl, který kopíruje NetFlow záznamy ze souboru vytvořeného kolektorem do databáze. Před použitím je nutné ve skriptu nastavit správné údaje pro přístup k databázi.
- `nftable.sql` - soubor obsahuje definici požadované struktury databázové tabulky. Tuto tabulku je nutné vytvořit na serveru MySQL ke kterému bude mít aplikace přístup.
- `detektor.conf` - hlavní konfigurační soubor aplikace. Před prvním spuštěním je nutné zde nastavit správné údaje pro přístup k databázi. Ostatní parametry které je možné nastavit jsou posány v tomto souboru.
- `detektor.jar` - vytvořená aplikace detektoru v jazyce java.
- `config/` - adresář ve kterém se nachází ukázkové konfigurační soubory pro jednotlivé testy. Kompletní seznam parametrů, které lze u těchto testů nastavit je popsán v těchto souborech.
- `nfdata/` - adresář s testovacími NetFlow daty pro bezpečnostní hrozby implementované v aplikaci

7.2 Používání programu

Před spuštěním aplikace je nutné naplnit databázi daty, na kterých chceme provádět test. Toho lze dosáhnout spuštěním skriptu `dbflow.pl`, kterému jako parametr předáme soubor s NetFlow daty, nebo unixový socket do kterého zapisuje Flowd kolektor:

```
./dbflow.pl datovy_soubor
```

Pokud již máme databázi naplněnu daty, je možné spustit aplikaci příkazem:

```
java -jar detektor.jar konfiguracni_soubor
```

Jediným parametrem je cesta ke konfiguračnímu souboru aplikace. Pokud tento parametr není uveden použije se jako konfigurační soubor `detektor.conf` v aktuálním adresáři, pokud existuje.

Kapitola 8

Závěr

Tato diplomová práce si kladla za cíl navrhnout a implementovat detektor bezpečnostních hrozeb v síťovém provzu, který by jako podklad pro svoji práci využíval záznamy o síťovém provozu získávané pomocí technologie Cisco NetFlow.

V úvodní části práce je především představena technologie NetFlow a možnosti jejího využití k detekci síťových útoků a detekci bezpečnostních hrozeb. Ukazuje se, že tato technologie je pro monitorování síťového provozu velmi perspektviní a v případě použití akcelerovaných HW NetFlow sond je použitelná i pro tvorbu detailních statistik pro gigabitové a desetigigabitové datové linky.

V dalším kroku jsem provedl zmapování běžných bezpečnostních hrozeb a útoků, vyskytujících se především v prostředí internetu, z pohledu jejich možné detekovatelnosti na síťové a transportní vrstvě síťového modelu. Několik z těchto hrozeb jsem následně v další části této práce detailně analyzoval, s cílem odhalit jejich charakteristické vlastonsti, kterými se projevují v záznamech o síťové komunikaci NetFlow.

Na základě těchto poznatků jsem implementoval aplikaci detekující v NetFlow záznamech tyto případy: hromadné rozesílání nevyžádané pošty, útok na SSH server slovníkovou metodou nebo metodou hrubé síly, detekce nalezení VoIP hovorů a detekce použití peer-to-peer sítí pro přenost souborů.

Funkčnost této aplikace byla úspěšně ověřena na vzorových datech pro jednotlivé případy i na datech pocházejících z reálného provozu celé počítačové sítě za delší období.

Tato prácé může být přínosem pro správce počítačových sítí různých velikostí nebo pro poskytovatele internetového připojení, kde detekované údaje mohou být použity pro zvýšení bezpečnosti a spolehlivosti dané sítě.

Možnosti rozšíření a dalšího vývoje projektu jsou blíže zhodnoceny v kapitole 6.3. Jedná se především o možnost doplnění dalších testů pro detekci ostatních bezpečnostních hrozeb, které nebyly v této práci implementovány.

Literatura

[1] Caligare s.r.o.: NetFlow Portal - Netflow export format. `http://netflow.caligare.com/`.

[2] Cisco Systems Inc.: Cisco IOS NetFlow. `http://www.cisco.com/go/netflow`.

[3] Krčmář, P.: *Linux - tipy a triky pro bezpečnost*. Grada Publishing, a.s., 2004, iSBN 80-247-0812-4.

[4] RFC 3551: RTP Profile for Audio and Video Conferences with Minimal Control. (2003).

[5] RFC 3954: Cisco Systems NetFlow Services Export Version 9. (2004).

[6] WWW stránky: MySQL. `http://cs.wikipedia.org/wiki/MySQL`, (duben 2010).

[7] WWW stránky: Netflow. `http://cs.wikipedia.org/wiki/Netflow`, (prosinec 2009).

[8] WWW stránky: Spam. `http://cs.wikipedia.org/wiki/Spam`, (prosinec 2009).