VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZABEZPEČENÍ DATOVÉ SÍTĚ S VYUŽITÍM NETFLOW DAT

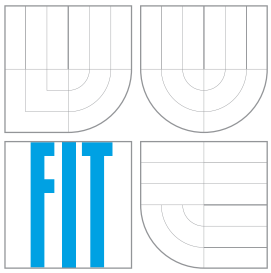
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK CZUDEK

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

ZABEZPEČENÍ DATOVÉ SÍTĚ S VYUŽITÍM NETFLOW DAT

NETWORK PROTECTION USING NETFLOW DATA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK CZUDEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MARTIN ŽÁDNÍK

BRNO 2010

Abstrakt

Tato práce se zabývá možností lepšího zabezpečení datové sítě na základě NetFlow protokolu. Konkrétně se jedná o detekci síťového skenování na základě předem daných pravidel vyhledávání této anomálie v NetFlow datech. Další částí je možnost retrospektivní analýzy získaných dat, a tím dosažení přesnější detekce útoků v dané síti. V rámci této bakalářské práce je navržena aplikace, která využívá předem daná pravidla pro detekci skenování a následně vyhledává toky směřující na porty chráněné touto aplikací, a zpětně je porovnává z detekovanými skeny. Tímto způsobem je dosažená přesnější detekce útoků.

Abstract

This thesis deals with the possibility of greater security of network based on NetFlow protocol. Specifically, the detecting network scans based on predefined rules to found this anomaly in the NetFlow data. Next part is the possibility of retrospective data analysis, thereby achieving more accurate detection of attacks on the network. In this work designed application uses predetermined rules to detect the scans and then looks the flows towards the ports witch are protected by the application and than compares with detected scans. In this way, more accurate detection of attacks is achieved.

Klíčová slova

NetFlow, IDS, skenování, detekce útoků, Timemachine, zabezpečení síťě

Keywords

NetFlow, IDS, scanning, Detection of attacks, Timemachine, Network Security

Citace

Marek Czudek: Zabezpečení datové sítě s využitím NetFlow dat, bakalářská práce, Brno, FIT VUT v Brně, 2010

Zabezpečení datové sítě s využitím NetFlow dat

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Martina Žádníka

. . . . . .
Marek Czudek
9. května 2010

Poděkování

Chtěl bych v první řadě poděkovat panu Ing. Martinu Žádníkovi za odbornou pomoc a mým nejbližším, kteří mi byli oporou.

Obsah

Kapitola 1

Úvod

Internet se v dnešní době stává čím dál tím populárnější a stává se rovněž nedílnou součástí každodenního života. S rozšiřující se popularitou Internetu roste rovněž nesčetně mnoho služeb, aplikací a portálů na něm provozovaných. Tyto služby nám poskytují mnoho možností jak komunikovat se svými blízkými a kamarády kdekoli jsou a to velice jednoduše a levně. Můžeme taky sdílet své fotografie a spoustu jiných věcí s kýmkoli chceme. Portály, jako jsou různé encyklopedie, nám poskytují informace k téměř každému oboru, o kterém bychom se rádi něco dozvěděli. Rovněž máme k dispozici různé televizní kanály, kde můžeme sledovat, co se nám zachce. Co se týče zábavy nám internet poskytuje možnost jako je herní multiplayer, kde lidé z různých koutů světa mohou vstoupit do virtuálních her a užívat si společně zábavu, při které mohou samozřejmě taky komunikovat.

Nedílnou součástí Internetu je rovněž rozmáhající se internetové bankovnictví, které nám poskytuje možnost spravovat své finance kdykoliv a kdekoliv jsme. Máme rovněž nad těmito operacemi větší přehled. Mnoho firem v dnešní době rovněž využívá internetové obchody tzv. e-shopy, které dávají zákazníkovi možnost prohlížení si zboží dané firmy přímo na webu a jeho okamžitého objednání.

Internet se stal v dnešní době součástí naší společnosti. Všechny služby jako je nakupování, získávání informací, bankovnictví, vydělávání a mnoho mnoho dalších možností našlo místo právě v této oblasti. Bohužel tyto služby našly i své protivníky, kteří je chtějí buď zneužívat, nebo dokonce zničit. Někteří to dělají pro peníze, někteří jenom tak pro zábavu a další proto, aby si získali respekt.

Kvůli těmto hrozbám se začaly vyvíjet různé systémy a algoritmy, které by byly schopny tyto lidi zastavit a znemožnit jim zneužití těchto služeb nebo jejich poškození. Jedním z takovýchto prostředků vyvinula firma Cisco Systems. Tato firma vyvinula protokol zvaný NetFlow, který je využíván pro monitorování IP toků na síti.

1.1 Netflow

NetFlow je otevřený protokol vyvinutý firmou Cisco Systems. Tento protokol byl původně určený pouze pro potřeby firmy. Protokol slouží k monitorování síťě na základě IP toků, což umožňuje administrátorům získávat v reálném čase informace o provozu na síti. Tyto informace mohou sloužit jak pro detekci útoků na síti, tak rovněž pro ISP (Internet Service Provider) k účtování konektivity [2, 1].

1.1.1 Úvod do NetFlow

Nejdůležitějším prvkem v NetFlow je IP tok (ang. „flow“). IP tok je alfou a omegou celé NetFlow technologie.

Definice toku. IP tok je definován jako sekvence paketů, které sdílí následující vlastnosti.

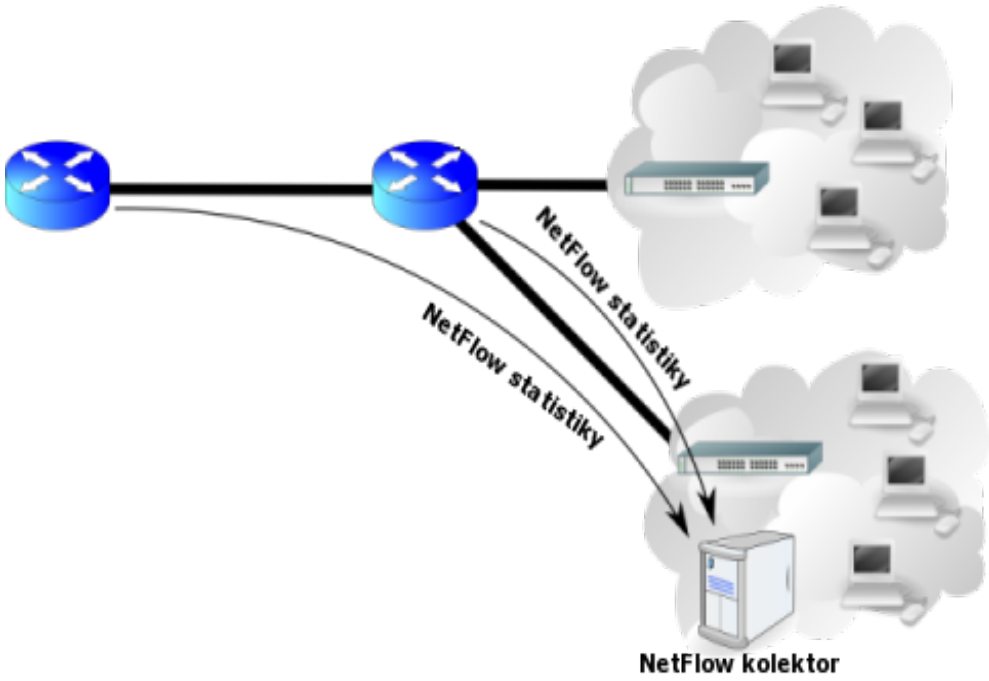
- Zdrojová IP adresa
- Cílová IP adresa
- Zdrojový port
- Cílový port
- Číslo protokolu
- Rozhraní (interface)
- Typ služby (Type of Service)

Pro každý IP tok je rovněž zaznamenávána doba vzniku toku, délka jeho trvání, počet přenesených paketů a bajtů a další údaje. Získávané údaje záleží na verzi NetFlow protokolu. Nejvíce rozšířenou verzí je NetFlow verze 5. Ta definuje pětici, dobu vzniku toku, dobu trvání, počet přenesených paketů a bajtů a TCP příznaky [2].
Nyní představím princip fungování:

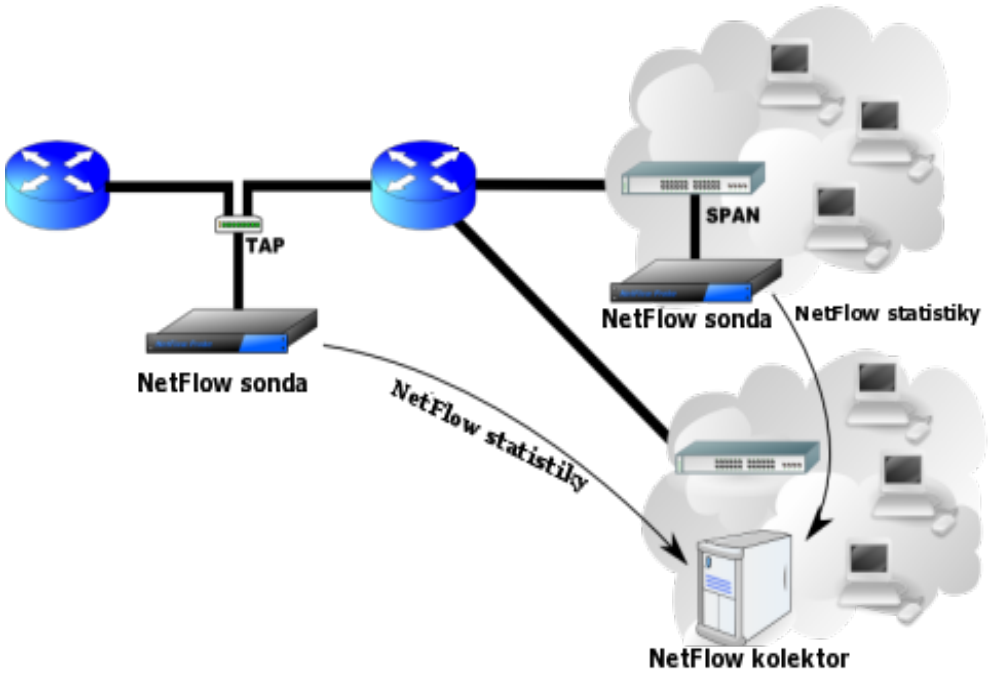
1. **Monitorování a export** - nejprve je potřeba získat informace o tocích. O získávání těchto informací se stará tzv. exportér. Nejčastěji se jedná o různá síťová zařízení jako jsou routery nebo switche, viz Obrázek 1.1. Exportéry mohou být rovněž specializovaná zařízení, kterým se říká NetFlow sondy, viz Obrázek 1.2. Tyto exportéry monitorují toky a nasledně odesílají nasbírané informace do kolektoru.

2. **Sběr dat** - kolektor je prvek, který přijímá data odeslané exportérem. Kolektor dokáže jak filtrovat, tak agregovat přijatá data. Může rovněž přijímat od více než jednoho exportéru. Nasbírané informace nejčastěji ukládá do databáze.

3. **Zpracování a zobrazení dat** - takto přijatá data můžeme analyzovat a následně zobrazovat v podobě různých grafů. Dále můžeme získávat různé informace a parametry síťového provozu.



Obrázek 1.1: NetFlow tradiční architektura [2]



Obrázek 1.2: NetFlow moderní architektura [2]

1.1.2 Netflow záznam

NetFlow záznam, viz Obrázek 1.3, obsahuje pouze data týkající se toku, neobsahuje žádná data, které se nacházejí v přenášených, monitorovaných paketech. Velikost NetFlow záznamu je tedy nesrovnatelně menší než data přenesená v rámci daného toku.

Obrázek 1.3: NetFlow záznam [2]

Toto je způsobeno tím, že důležité a podstatné informace o konkrétním toku se pouze agregují do jednoho záznamu. V záznamu se pouze zvětšuje počet přenesených paketů a bajtů atp. Také TCP příznaky se agregují tak, že v záznamu jsou zaznamenány příznaky, které se vyskytly v průběhu celého toku. Takovýmto způsobem je možné uchovávat záznamy o velkém množství toků, které jsou uloženy na kolektoru [2].
Informace o toku jsou v NetFlow protokolu závislé na jeho verzi. V paketu NetFlow verze 5 jsou obsaženy následující položky [1]:

- Číslo verze
- Sekvenční číslo
- SNMP index vstupního a výstupního rozhraní
- Čas začátku a konce IP toku
- Počet bajtů a paketů v toku
- Údaje z L3 hlavičky:
  - Zdrojové a cílové IP adresy
  - Zdrojové a cílové porty
  - IP protokol
  - Type of Service (ToS)
- U TCP toků obsahuje množinu tvořenou sjednocením všem TCP flagů, které se v toku vyskytly.
- Směrovací informace:
  - IP adresa přístího hopu
  - Maska cílové a zdrojové IP adresy

1.1.3 Využití NetFlow

Zde je stručný seznam možností využití NetFlow protokolu

- **Monitorování sítí**
- **Monitorování aplikací**
- **Monitorování uživatelů**
- **Plánování sítí**
- **Bezpečnostní analýza**
- **Vyúčtování provozu**
- **Ukládání NetFlow dat a Data Mining**

1.1.4 NetFlow nástroje

Zde bych chtěl představit pár nástrojů pro práci s NetFlow protokolem. Můžeme najít opravdu hodně open-source nástrojů, ale většina z nich podporuje pouze práci s NetFlow v5. Jedním z kvalitních open-source řešení je sada NFDUMP tools [12].

- **nfcapd (netflow capture daemon)** - tento nástroj se chová jako kolektor, který sbírá NetFlow data ze síťě a následně je ukládá do souboru. Defaultně ukládá pě-timinutové úseky, poté soubor uzavře a otevře nový.
- **nfdump (netflow dump)** - tento nástroj se využívá pro zobrazení dat z nfcapd. Pomocí něho můžeme filtrovat zabrazovaná data a rovněž generovat top N statistiky. Využívá podobnou syntaxi jako tcpdump.
- **nfprofile (netflow profiler)** - tento nástroj se využívá pro filtrování dat z nfcapd a jejich následné uložení do souboru pro další zpracování.
- **nfreplay (netflow replay)** - tento nástroj se využívá pro přeposlání dat z nfcapd po síti jinému kolektoru
- Dalšími nástroji mohou být **NfSen**, který je graficou nadstavbou nad NFDUMP tools, nebo **fprobe**, který je softwarovým řešením NetFlow sondy
- A další

Kapitola 2

Typy útoků na síti

Typy útoku na síti můžeme rozdělit na dva základní druhy. Jedním je využití objemových dat, které mají za úkol zahltit zdroje oběti, a tím dosáhnout pádu služby. Tímto způsobem tedy útočník znepřístupní službu legitimním uživatelům. Těmto útokům se říká Denial of Service (DoS) útoky, které v krátké době získaly velkou popularitu. Dalším druhem jsou pakety se speciálním obsahem, které využívají slabá místa napadaného systému. Útočník chce nejdříve získat přístup do systému a následně k datům, které daný systém obsahuje.

2.1 IP Spoofing

Tato technika umožňuje útočníkům provádět útoky tak, aby je nebylo možné jednoduše identifikovat. Principem je přepsání hlavičky odchozích paketů. Konkrétně se přepisuje vlastní, tedy IP adresa zdroje. Při zachování podmínek IP adresace je možno nahradit vlastní IP adresu téměř jakoukoliv. Na cílovém systému je sice možno v protokolech zjistit připojení útočníka a případně dohledat jeho činnost v systému, avšak jako adresa bodu, odkud se útočník připojuje, je právě ta IP adresa, kterou útočník podvrhl. Pokud útočník použije IP adresu stroje, který je v okamžiku jeho připojení k cílovému systému nedostupný, může být takové připojení detekováno v protokolech na cílovém systému jako chyba. Správce tohoto systému tak může jednoduše odhalit pokusy o připojení klienta, který uvádí nekorektní zdrojovou IP adresu. Tento způsob maskování není použitelný tam, kde útočník potřebuje zachytávat odpovědi cílového systému. Navázání obousměrné komunikace je nemožné. Nic však nebrání v zasílání instrukcí cílovému systému [9, 5].

2.2 ARP Spoofing

ARP (Adress Resolution Protocol) Spoofing využívají útočníci v místních sítích, kde se mohou vydávat za jiný počítač. Pokud počítač v místní síti chce poslat data jinému počítači, u kterého zná pouze IP adresu, tak protokolem ARP odešle všem uzlům v síti dotaz, který se zeptá „Kdo má tuto IP adresu, nechť mi pošle svoji MAC adresu“. Tento útok tedy spočívá v neustálem zasílání odpovědí s MAC adresou, který způsobí, že si cíl zaznamená falešnou MAC adresu do svých vnitřních tabulek, a data bude posílat na ni. Pokud je cílem útočníka odposlouchávat komunikaci mezi dvěma stroji, stačí mu pouze oběma podstrčit svoji MAC adresu a přijatá data posílat skutečným adresátům.

2.3 Denial of Service a Distributed Denial of Service

Denial of Service útoky charakterizuje blokování služby sítě zaplavováním spojení, zhroucením serverů nebo programů běžících na serverech, vyčerpáváním zdrojů na serveru nebo jinak brání legitimním klientům v přístupu ke službám sítě. Tento útok je prováděn z jediného počítače.

Denial of Service útoky mohou mít více podob, od útoku jednoho paketu, který způsobí zhroucení serveru, až po koordinované záplavy paketů od mnoha botnetů. Při útoku jednoho paketu je poslán do sítě pečlivě přizpůsobený paket, který využívá známé zranitelnosti operačního systému nebo aplikace a zablokuje server, nebo některé služby jím poskytované [10].

Útoky těchto typů, které budou nyní blíže představeny, čerpají informace z následujících zdrojů [10, 14, 6, 3].

2.3.1 Distributed Denial of Service

Distributed Denial of Service rozvíjí mechanizmy, které jsou čím dál tím sofistikovanější a účinnější pro dosažení cíle. Tyto mechanizmy využívají záplavové útoky. Zdroje na serveru nebo na síti jsou narušeny nebo vyčerpány záplavou paketů. Po napadení z jednoho místa může být záplava celkem snadno identifikována a izolována.

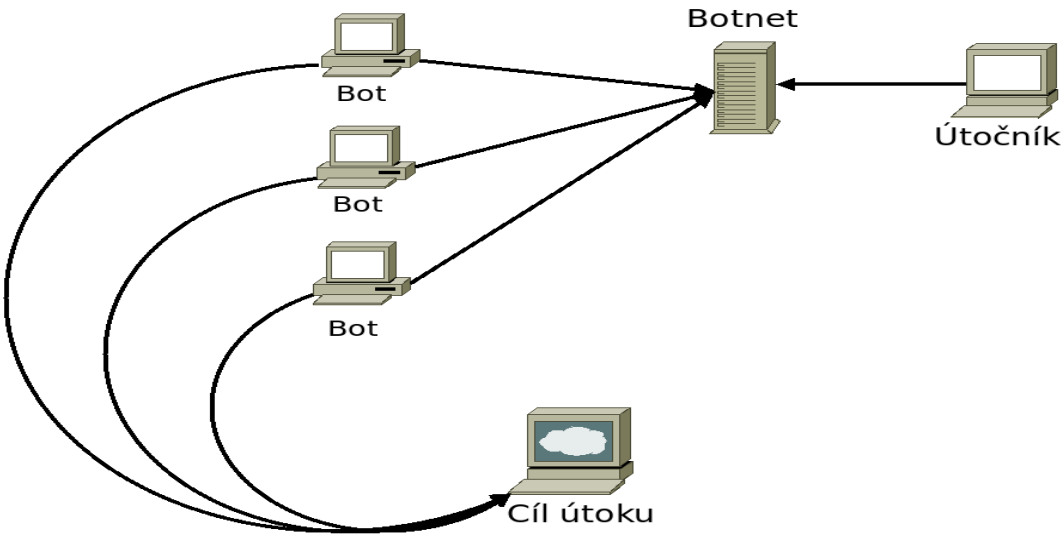
Při DDoS útoku se ale většinou využívá k zasažení cíle velkého množství počítačů. Některé útoky jsou prováděny jednoduchým způsobem, jako třeba poslání nekonečného proudu dat k zaplavení síťových spojení na serveru. Jiné útoky, jako je třeba SYN útok, používají pečlivě upravené pakety k vyčerpání zdrojů za účelem zabránit legitimním klientům v připojení k serveru, viz Obrázek 2.1.

Bez ohledu na druh útoku se k DDoS používá větší množství počítačů (botnetů) s koordinovaným přístupem. Tyto přístroje, známé jako botneti, musí být předtím kompromitovány speciálním kódem, který dovolí útočníkovi tento počítač kontrolovat. Posíláním příkazů těmto botnetům přes skryté komunikační kanály mohou útočníci provádět rozsáhlé koordinované útoky. Protože útok pochází od velkého množství počítačů rozmístěných v síti, je bohužel velice obtížné tyto útoky identifikovat a izolovat. V mnoha případech je velice obtížné oddělit legitimní provoz od útočného, viz Obrázek 2.1.

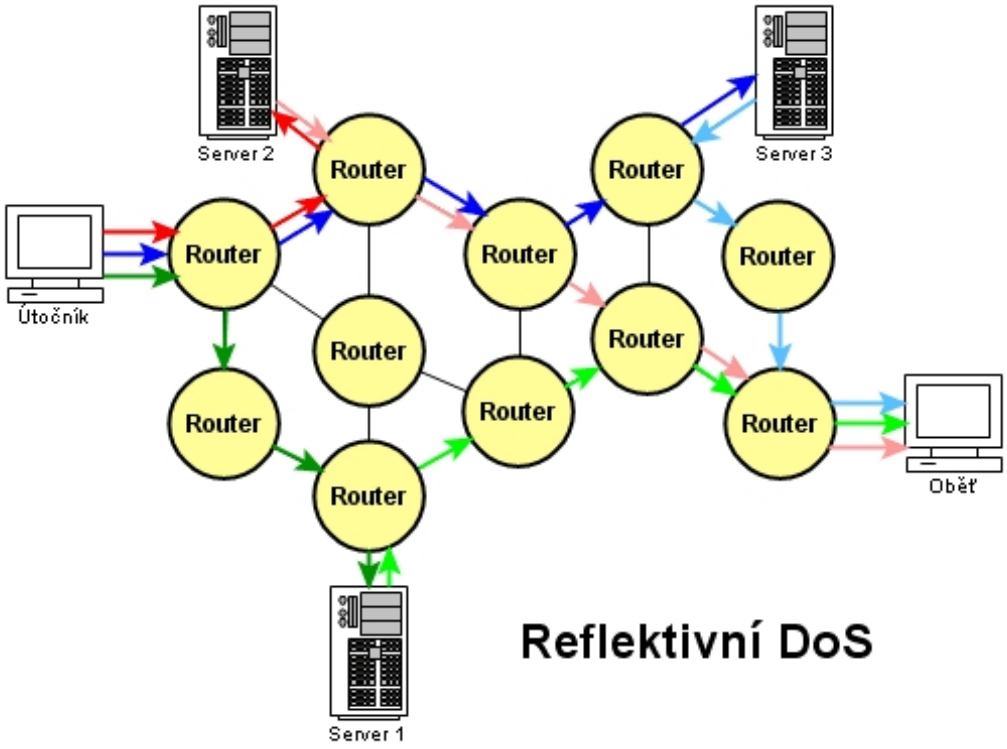
2.3.2 Distributed reflection Denial of Service

Jedná se o útoky, které jsou podobné předchozím technikám, ovšem k útoku používají jiné počítače (nebo routery) jako prostředníky. Tito prostředníci nemusí být kompromitováni, to znamená, že útočník je nejdřív nepotřebuje napadnout. Tyto útoky se provádí hlavně distribuovaně.

Největší výhodou tohoto útoku je minimální možnost vystopování útočníka. Data (použitá k zahlcení) se totiž nepřenášejí stále stejnou cestou, jelikož se při útoku mění počítače, od kterých se útok „odráží“ (proto se jim říká reflektivní). Vyhledávání útočníka se provádí tak, že se od oběti postupuje postupně po routerech směrem k útočníkovi. Vždy se na routeru zjistí, ze kterého portu útok přichází, a toto se provede znovu na routeru, který je připojen k danému portu, a takto stále dokola. Z toho důvodu je velice obtížné vystopování útočníka, viz Obrázek 2.2.



Obrázek 2.1: Distributed Denial of Service [14]



Obrázek 2.2: Reflektivní DDoS útok [14]

2.3.3 TCP SYN flood

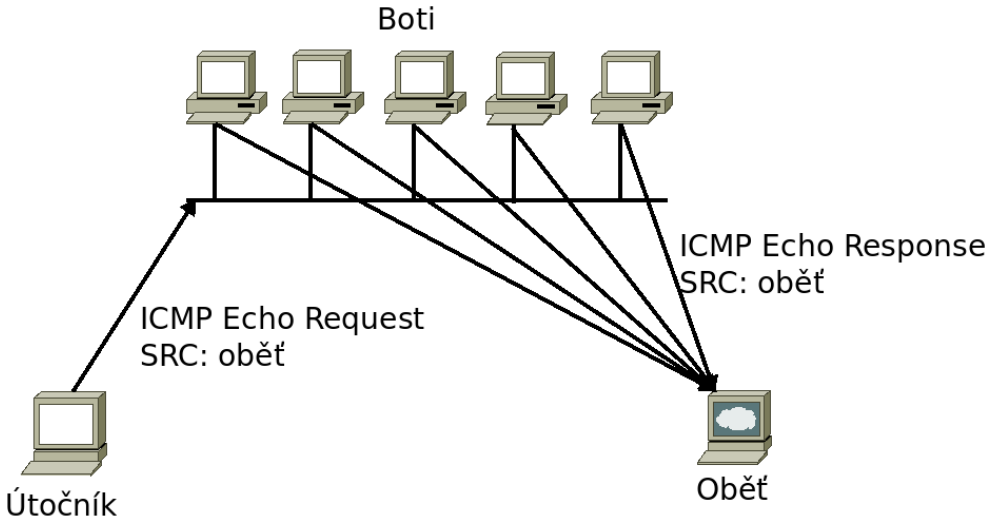
Útočník využívá slabá místa v architektuře TCP. Pro navázání spojení je třeba provést tzv. three-way handshake, kde klient zašle požadavek o spojení na server SYN, server odešle

odpověď SYN a ACK a poslední fází je, že klient potvrdí spojení pomocí ACK. Tímto způsobem se v TCP architektuře navazuje spojení. V tomto útoku, ale třetí fáze neproběhne a server stále čeká na dokončení synchronizace spojení. SYN-flood je známý způsob útoku, ale v moderních sítích obvykle neúspěšný. Funguje jedině tehdy, pokud server alokuje prostředky po obdržení paketu SYN ještě před tím, než obdržel paket ACK. Máme dva způsoby, jak zařídit, aby se server nedočkal paketu ACK. Útočník buď může zaslání paketu ACK opomenout, nebo může poslat paket SYN se špatně uvedenou IP adresou. Tomuto způsobu se říká IP spoofing , kdy klient nastrčí jako svoji IP-adresu nějakou jinou adresu a server pošle paket SYN-ACK na jinou IP adresu, a samozřejmě se paketu ACK nedočká. Pokud server přiděluje prostředky těmto napůl otevřeným spojením (například si informace ukládá do fronty a odtud je odebírá, když přijde paket ACK) a je zaplaven (flooded) podvodnými pakety, zanedlouho jsou prostředky vyčerpány. To může způsobovat problémy jako například zpomalení serveru, ale třeba i zhroucení systému a počítač musí být pak lokálně restartován.

2.3.4 Smurf attack

Smurf útok využívá chybnou konfiguraci systému, který dovolí rozeslání paketů všem počítačům v sítí pomocí broadcast adresy. Útok je proveden zasláním ICMP paketu `ECHO Request`, který má upravenou zdrojovou IP adresu a je odeslán na všechny počítače v síti pomocí broadcast adresy. Díky upravené zdrojové IP adrese všechny počítače odpoví na tento ICMP paket `ECHO Request` ne na skutečnou zdrojovou IP adresu, ze které byl tento dotaz odeslán, ale na upravenou IP adresu, na níž je tento DoS útok nasměrován.

Díky této technice je možné odesláním jenoho paketu vygenerovat několikanásobně více paketů, čímž se výrazně zvýší síťový provoz a způsobí nedostupnost napadeného počítače, případně celé podsítě, viz Obrázek 2.3.



Obrázek 2.3: Smurf útok [14]

2.3.5 Ping flood

Ping flood je útok hrubou silou. Aby tento útok mohl být použitý a byl úspěšný, tak se předpokládá, že útočník má k dispozici rychlejší připojení do sítě než jeho oběť. Tento

útok je proveden zasláním dat na adresu oběti, čímž dojde k vyčerpání přenosové kapacity směrem k oběti. Takto provedený útok způsobí nedostupnost tohoto počítače pro ostatní uživatelé. Takto koncipovaný útok je možné provést běžným systémovým nástrojem jako je `ping`.

2.3.6 HTTP flood

Webové služby jsou v dnešní době asi nejvíce používanou internetovou službou na sítí. Tento protokol běží na portu 80, který je otevřen téměř na každém počítači. Útočníci tedy využívají botnetů na posílání mnoha http dotazů. Tyto dotazy mohou směřovat na stahování velkých souborů ze serveru, což může způsobit, že server, který tato data čte z disku, ukládá do paměti a následně dělí na pakety, při takovémto množství dotazů vyčerpá své zdroje a zhroutí se. Těmto útokům je téměř nemožné se bránit. Často se rovněž stává, že pády těchto serverů jsou způsobeny tím, že některý z velkých serverů např. BBC odkáže své uživatele na nějaký soubor na malém serveru a ten nezvládne obsloužit tolik příchozích požadavků.

2.4 Hádání hesla

Zde bych chtěl uvést metody, které se dají použít pro získání přístupu do systému pomocí uhádnutí hesla uživatele. Metody, které bych rád přiblížil jsou *„brute force“* metoda (*„metoda hrubou silou“*) a *„slovníkový útok“*. Pro využití těchto technik potřebujeme rovněž nějaký specializovaný program, pomocí kterého budeme tyto útoky provádět.

2.4.1 Brute force útok

Útok hrubou silou spočívá v otestování všech možných kombinací (slov) nad zvolenou množinou znaky resp. abecedou, která může mít různá omezení. Zde si musíme uvědomit, jaký má dopad délka hesla a velikost použité množiny na počet všech možných variant. Pokud vytvoříme heslo o n znacích a použitá množina bude obsahovat m znaků, pak všech možných slov o délce n bude m^n. Jako příklad můžeme uvést použití šestipísmenného hesla nad anglickou abecedou malých písmen což nám dá 26^6, a to je přibližně 308 miliónů. Pokud použijeme osmi znakové heslo a rozšíříme množinu použitých znaků na abecedu s malými i velkými písmeny, pak všech možných kombinací je 62^8 což je zhruba sedm set tisíckrát více variant. Útočník k tomu často nezná délku hesla, proto mu nezbývá nic jiného, než postupně vyzkoušet všechny jednopísmenné až (n-1)-písmenné varianty. Proto se tomuto útoků říká útok hrubou silou. Tento útok není nijak efektivní.

2.4.2 Slovníkový útok

Slovníkový útok je oproti brute force útoku úspěšnější tam, kde uživatelé spoléhají na jednoduchá a snadno zapamatovatelná hesla. Proto většinou tito uživatelé volí běžná slova, která lze najít v běžných slovnících. Tento útok se rovněž od brute foce útoku liší tím, že má neporovnatelně menší počet variant pro uhádnutí hesla. Varianty jsou tedy obsaženy v externím souboru, který je používán konkrétním programem. Tyto soubory (slovníky) můžeme najít všude možně na internetu, nebo si taky můžeme takový slovník vytvořit sami.

2.5 Skenování

Skenování je jedním z nejzákladnějších útoků na síti, které předchází většině zneužití sítě. Tímto způsobem může útočník zmapovat síť a všechny stanice se zde nacházející. Cílem je zjistit informace o těchto stanicích pro následné jejich využití při dalším útoků, kterým může být např. DoS nebo jiné. Při skenování se používají techniky jako jsou:

- **Skenování portů** - touto technikou zjišťujeme dostupné služby, které běží na konkrétní stanici
- **Skenování IP adres** - touto technikou zjišťujeme, které stanice jsou v dané podsíti dostupné

Skenování sítě také dělíme na tři typy. Každý typ používá jiný způsob, kterým skenuje stanice. Pro skenování lze použít jak protokol TCP, tak rovněž protokol UDP. Porty, které jsou skenovány se mohou nacházet v několika stavech:

- **open, accepted** - pokud je port otevřený a je možno se službou běžící na tomto portu navázat spojení
- **closed, denied** - pokud je port uzavřený. Tím pádem při pokusu o navázání spojení na tento port u TCP je zpětně zaslán paket s příznaky RST a ACK. Při UDP je zpětně zaslán ICMP paket s typu 3 (unreachable).
- **filtered, blocked** - tento stav označuje, že na portu nebyla zjištěna žádná odpověď

Zde jsou ve zkratce popsány všechny skenovací techniky, které se mohou používat zvlášť nebo je můžeme také kombinovat. Tyto techniky se vyznačují odlišným cílem, kterého chceme dosáhnout. Buď chceme zjistit otevřené porty na jedné stanici a služby na ní běžící nebo chceme zjistit dostupnost konkrétního portu (služby) na více než jedné stanici. Můžeme ale tyto cíle skloubit dohromady a zjistit dostupnost několika portů (služeb) na více stanicích v dané síti.

- **Vertikální skenování** - zde útočník skenuje více portů na jedné stanici
- **Horizontální skenování** - zde útočník skenuje jeden port na více stanicích v podsíti
- **Blokové skenování** - zde útočník používá kombinací obou výše uvedených technik

2.5.1 Skenovací techniky

Zde více rozepíšu jednotlivé skenovací techniky, které jsou nejčastěji používány jak protokolem TCP tak UDP.

U TCP protokolu můžeme využít několik technik. To vychází z toho, že TCP protokol využívá různá nastavení příznaků v paketu a taky se navazuje a ustanovuje spojení. Při navazování spojení probíhá tzv. three-way handshake. Při otevřeném portu je zpětně zaslán paketu s nastavenými příznaky SYN a ACK. Pokud je port uzavřený tak je zaslán zpětně paket s příznakem RST, a pokud je port filtrován, tak žádná zpětná odpověď není zaslána.

U UDP protokolu se využívá pouze zaslání UDP paketu s hlavičkou, který neobsahuje data. Pokud je skenovaný port otevřený jsou zpětně zaslána nějaká data. Pokud je port uzavřen je zpětně zaslán ICMP paket typu 3. A pokud je port filtrován je zpětně zaslán nějaký ICMP paket typu (1, 2, 9, 10 nebo 13).

SYN skenování

Při této technice skenování se posílá pouze paket s příznakem SYN a tím pádem se dosáhne toho, že kompletní navázání spojení TCP se nedokončí. Posláním pouze paketu s příznakem SYN se docílí toho, že skenovaný port nám odpoví při otevřeném portu paketem s příznaky SYN a ACK. Nyní by útočník podle three-way handshake měl zaslat paket s příznakem ACK, ale toto útočník neudělá, a proto spojení není navázáno. Výhodou této techniky je, že s největší pravděpodobností nebude kompletně navázáné spojení logováno.

Connect skenování

Tato technika využívá kompletního navázání spojení. Pokud skenovaný port je otevřený tak proběhne kompletní navázání TCP spojení. Tato technika nám ale poskytuje stejné informace, které nám může poskytnout SYN skenování, a proto je tato technika velice zřídka využívána. Další nevýhodou této techniky je, že kompletně navázána spojení jsou logována.

Null, Xmass a FIN skenování

TCP protokol definuje, že pokud je port uzavřený, tak na každá příchozí data neobsahující RST příznak odpoví paketem s příznakem RST. Pokud je port otevřený, tak na každá příchozí data, kde nejsou v paketu nastaveny žádné z příznaků SYN, RST nebo ACK reaguje zahozením příchozího paketu.

- **Xmass skenování** - využívá nastavení příznaků FIN, URG a PSH
- **Null skenování** - využívá poslání paketů bez nastavení příznaků
- **FIN skenování** - využívá nastavení pouze příznaku FIN

2.5.2 Skenovací nástroje

Pro využití těchto technik, kromě connect skenování, který je normálním navázáním spojení, je třeba použít speciální program. Tyto programy využívají knihovny pro práci s pakety a vyžadují administrátorská oprávnění. Nejvíce používaným nástrojem pro skenování je nmap (Network Mapper). Dalším využívaným nástrojem je Nessus.

Nmap

Network Mapper je open source program, který slouží k prozkoumávání síťě, ověření bezpečnosti počítače nebo pro zjištění služeb nabízených počítači v dané síti. Využívá se hlavně k zjištění otevřených portů na jednom nebo více počítačích v dané síti. Rovněž slouží k identifikaci služby běžící na konkrétním portu či zjištění operačního systému běžícího na daném počítači [13].

Nessus

Nessus je nástroj sloužící ke kompletnímu skenování stroje. Tento nástroj neslouží pouze k ověření, zda-li jsou skenované porty na daném stroji otevřené či nikoliv, ale hlavně zjišťuje bezpečnostní chyby služeb běžících na daných portech. Je tedy jakousi kombinací nástroje nmap a databáze známých bezpečnostních chyb konkrétních služeb [11]. Nessus se skládá ze dvou částí:

- klient - nessus
- server - nessusd

Před zahájením skenování se klient nejprve připojí k serveru, a tím je skenování prováděno ze stroje, na kterém je tato serverová část nainstalována. To má určité výhody. Ulehčení vašemu internetovému spojení. Pokud jste za firewallem, pak by mohl být problém protlačit skenování skrz, protože firewall může některé testy blokovat. Proto je dobré mít server umístěn mimo firewall, což nám ulehčí práci [11].

Kapitola 3

Detekce útoků na síti

Mechanismy pro detekci útoků jsou vyvíjeny tak, aby v co nejkratším čase s co největším úspěchem dokázaly detekovat útok v reálném čase. Pro detekci útoků využíváme několik strategii.

Jednou ze strategií je vyhledávání signatur v datovém toku [16]. Tyto signatury jsou uloženy v centrální databázi, která musí být čím jak nejlíp aktualizována. Tento způsob detekce je výhodný v tom, že je rychlý a nedochází k mylné detekci korektního toku. Jeho nevýhoda spočívá v tom, že tímto způsobem nejsme schopni detekovat nové neznámé útoky, které stále přicházejí [17].

Další strategii je detekce anomálií na síti. Touto strategii sledujeme normální chování systému, který očekává normální transakce a normální provoz. V tomto systému se využívá tzv. práh. Tento práh určuje normální chování systému. Pokud dojde do situace, že tento práh je překročen, systém detekuje anomálií a upozorní administrátora. Pokud je tento práh příliš nízký, dochází tak k mnoha negativním výsledkům. Proto můžeme tento práh zvyšovat, čímž docílíme větší citlivosti systému [17].

Třetí strategii je hybridní detekce, která kombinuje dvě předchozí metody. Detekce anomálií se využívají k získání nových signatur do databáze. Často tyto systémy bývají plně automatické, proto třeba dávat pozor, aby nedocházelo k vyhodnocení normálního chování jako detekce útoku. Tímto by mohlo dojít k vlastnímu shození služby.

Pro detekci útoků na síti se používají dva systémy. Jedním je Intrusion Detection systém (IDS)[15] a Intrusion Prevention system (IPS)[4].

3.1 Intrussion Detection System (IDS)

Toto je systém, který detekuje narušení (možný útok). IDS sleduje datové toky a hledá v nich pokusy o útok na konkrétní aplikace. Jedná se o pasivní zařízení, pouze sleduje, nezasahuje do provozu sítě. Prostřednictvím alertů a statistik poskytuje obsluze informace o útocích. IDS sonda se nejčastěji připojuje k HUBu nebo na zrcadlový port přepínače. Tento systém využívá výše zmíněné strategie detekce útoků [17, 15, 18].

3.2 Intrussion Protection System (IPS)

IPS je systém, který používá pro detekci útoků stejné strategie jako je IDS, ale na rozdíl od něj okamžitě reaguje. Tedy IPS se díky možnosti reagovat na útoky jeví jako daleko spolehlivější způsob ochrany. Bohužel jeho okamžitá reakce je často nepříliš vhodná pro

danou situaci. Tam kde IDS vyvolá pouze poplach a administrátor ho vzápětí vyhodnotí jako planý, tak IPS automaticky na útok zareaguje. Myslí si totiž, že útok je platný. Může tedy odpojit běžného uživatele nebo zcela zablokovat síťový provoz na daném síťovém segmentu [17, 4].

Některé IDS systémy dokáží zase spolupracovat s firewallem, který dynamicky mění svoji politiku tak, aby zamezil komunikaci vyhodnocené jako útok, a také reagoval na útok. IPS má ale opět nad IDS navrch. Je totiž navíc schopen útoku zabránit, ještě než je vůbec zahájena jakákoliv komunikace. Protože není závislý na firewallu a nemusí tak čekat, než firewall vykoná jeho požadavky.

3.3 Detekce útoků v NetFlow

Na základě NetFlow dat je možno dobře detekovat různá DoS a DDoS útoky. Mezi tyto útoky se řadí ICMP flood, UDP flood, DNS flood a taky TCP SYN flood. Velice dobře detekovatelné je rovněž skenování, kterým se budu následně více zabývat.

3.3.1 Skenování

Skenování je v NetFlow datech velice dobře rozpoznatelné. Díky tomu, že skenování probíhá velice rychle, tak se v NetFlow datech tvoří veliké množství toků ve směru ke skenovanému počítači, které obsahují nízký počet paketů a mají nastaveny různé příznaky, které byly nastavovány v průběhu celého toku. Tyto příznaky jsou závislé na použité skenovací technice. Rovněž se v těchto datech tvoří veliké množství toků, které směřují od skenovaného počítače k počítači, který skenování provádí. Takto získaná data můžeme tedy využít k detekci skenování počítačů na síti. Samozřejmě máme různé techniky skenování, a proto potřebujeme několik pravidel pro odhalení těchto možností skenování. Toto téma je obsaženo v kapitole návrhu a implementace aplikace 5.2.2. Pro demonstraci a ukázku skenování jsem použil oba výše zmíněné nástroje. Nástroj Nmap je jednoduchým a velice rychlým na použití. Co se týče nástroje Nessus, tak tam jsem musel nejdříve zprovoznit Nessus server (nessusd) a následně doinstalovat webového klienta. Vše bylo dostupné s pomocí home licence.

Nmap

Zde bych rád uvedl ukázku NetFlow dat získaných v době použití Nmap skeneru [13] s použitím SYN skenování, viz Obrázek 3.1. V těchto datech jsem vynechal sloupec s dobou trvání toku z důvodu větší přehlednosti. Doba trvání toku je stejně v případě sknování velice mizivá a v datech se zobrazuje jako nulová.

Date flow start	Proto	Src IP Addr	Src Pt	Dst IP Addr	Dst Pt	Flags	Packets	Bytes
2010-04-06 12:51:35.894	TCP	147.229.209.241	48567	147.229.176.14	758	....S.	1	44
2010-04-06 12:51:35.899	TCP	147.229.209.241	48567	147.229.176.14	1021	....S.	1	44
2010-04-06 12:51:37.440	TCP	147.229.209.241	48566	147.229.176.14	556	....S.	1	44
2010-04-06 12:51:37.441	TCP	147.229.209.241	48566	147.229.176.14	826	....S.	1	44
2010-04-06 12:51:37.441	TCP	147.229.176.14	826	147.229.209.241	48566	.A.R..	1	40

Obrázek 3.1: Skenování pomocí Nmap

Na této ukázce je možné vidět, že toky směřující od skenujicího počítače ke skenovanému mají nastaven pouze příznak SYN (....S.), z čehož můžeme usuzovat, že se opravdu jedná

o techniku SYN skenování. V důsledku odpovědi se vytvoří nový tok, který má nastaveny příznaky ACK a RST (.A.R..) a tím ukazuje, že skenovaný port 826 je uzavřený.

Nessus

Zde vidíme ukázku NetFlow dat, které jsme získali při použití nástroje Nessus [11] s využitím techniky SYN skenování, viz Obrázek 3.2. Data získáná pomocí Nmap, viz Obrázek 3.1, a pomocí Nessus jsou téměř identická. Liší se pouze ve velikosti zaslaného paketu. Zde máme tedy potvrzení toho, že nástroj Nessus využívá stejných principů a technik jako Nmap popsaných v kapitole 2.5.2.

Date flow start	Proto	Src IP Addr	Src Pt	Dst IP Addr	Dst Pt	Flags	Packets	Bytes
2010-04-06 12:46:37.954	TCP	147.229.209.241	51632	147.229.176.14	202	....S.	1	48
2010-04-06 12:46:37.954	TCP	147.229.176.14	202	147.229.209.241	51632	.A.R..	1	40
2010-04-06 12:46:37.954	TCP	147.229.209.241	35275	147.229.176.14	208	....S.	1	48
2010-04-06 12:46:37.946	TCP	147.229.209.241	48405	147.229.176.14	790	....S.	1	48
2010-04-06 12:46:37.954	TCP	147.229.209.241	24534	147.229.176.14	18	....S.	1	48

Obrázek 3.2: Skenování pomocí Nessus

Z tohoto úseku dat můžeme určit kdo koho skenoval, a jakým způsobem dostává odpověď na dotaz o dostupnosti konkrétního portu.

Kapitola 4

Time Machine

V mnoha případech může být velmi užitečné uchovávání informací o síťovém provozu na disku, abychom byli schopni se do těchto záznamů podívat pomocí retrospekce. Time Machine je systém, který byl vyvinut pro retrospektivní analýzu síťového provozu. Tento systém využívá tzv. „heavy-tail“ podstatu sítě, kde se odchytává pouze nejužitečnější provoz a ukládají se malé zlomky z celého provozu na síti. Time Machine je často využíván různými IDS systémy pro detekci útoků [7].

4.1 Úvod do Time Machine

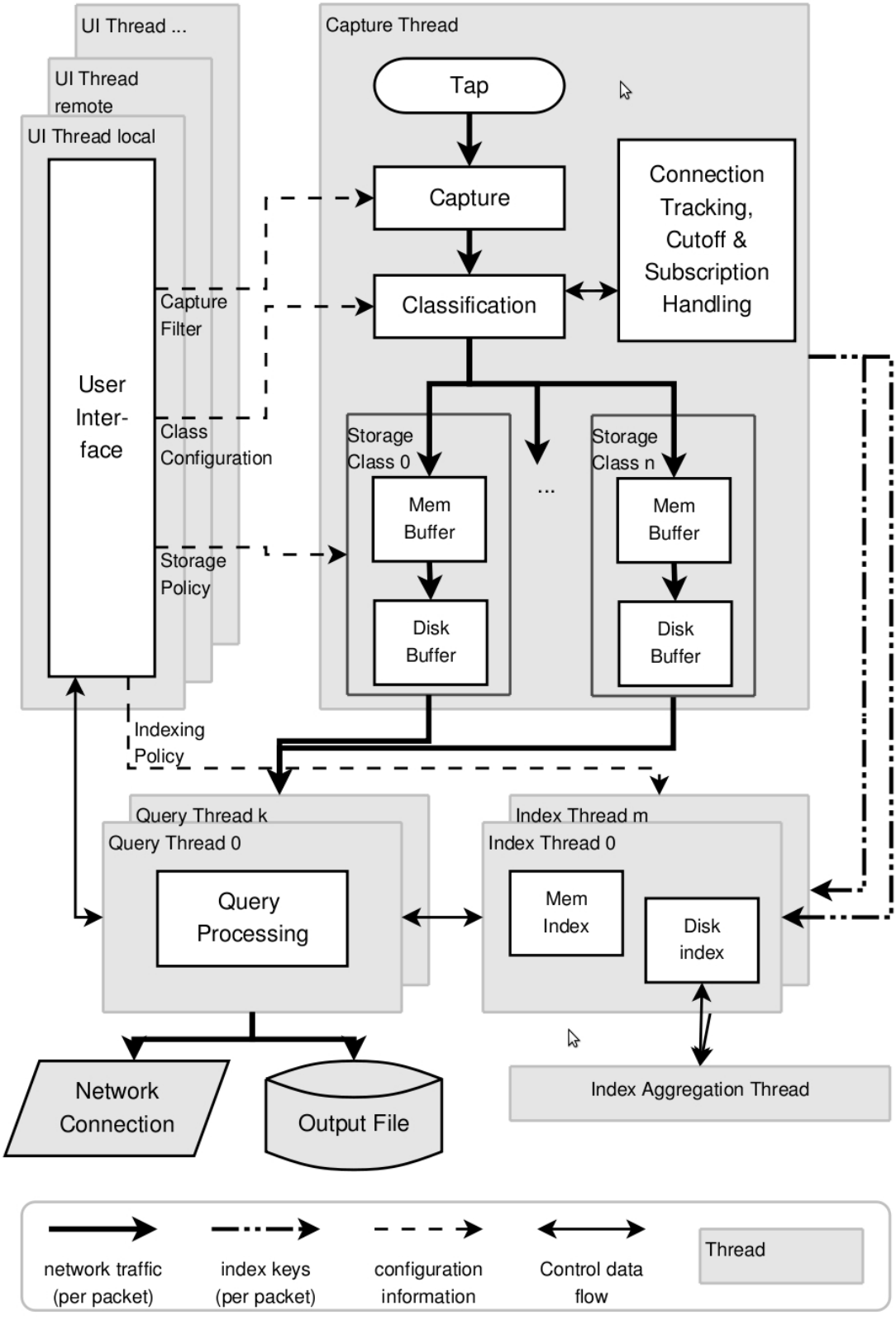
Celý systém pracuje podobně jak IDS systémy. Je nutné ukládat pouze důležitý provoz ze sítě tak, abychom byli schopni udržet co největší počet toků, které se vyskytly na síti. Pro dosažení tohoto je využíván tzv. „heavy-tail“, kde se pomocí *„cutoff“* odsekne zbytečná část toku. Ve většině případů je začátek toku tou nejdůležitější částí (obsahující protokol, handshakes, autentizační dialogy, atd.) [7]. Následně je možné takto získaná data analyzovat a klasifikovat. Time Machine nejčastěji využívá pro ukládání získaných dat databázi, ale je také možné využít ukládání do souborů.

Proto, abychom byli schopni plně využít možnosti Time Machine, je zapotřebí mít k dispozici i nějaký detekční systém (IDS). Mezi IDS a Time Machine je vytvořeno přístupové rozhraní, přes které IDS systém komunikuje s Time Machine. IDS tímto využívá retrospektivní analýzy získaných dat v reálném čase s daty získanými v minulosti.

4.2 Time Machine Design

Zde bych rád v bodech popsal design celého Time Machine, viz Obrázek 4.1, který se skládá z několika oddělených částí, které spolu komunikují.

- **Vlákno pro odchytávání paketů** (*Capture Thread*) - toto vlákno je zodpovědné za odchytávání paketů na síti, jejich klasifikaci, monitorování odseknutí a přiřazení paketů příslušnému skladovacímu místu.
- **Indexovací vlákna** (*Index Threads*) - tato vlákna udržují indexovací data, která poskytují dál dotazovacím vláknům.
- **Dotazovací vlákna** (*Query Threads*) - tato vlákna jsou schopna účinně lokalizovat a získat potřebná data, která jsou uložena v paměti nebo na disku.

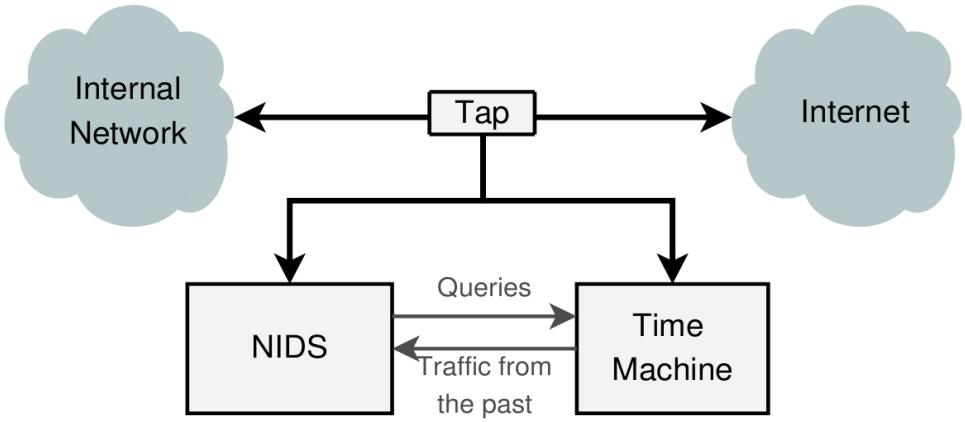


Obrázek 4.1: Time Machine [7]

- **Agregační indexovací vlákno** (*Index Aggregation Thread*) - toto je doplňkové vlákno, které uchovává indexovací data souborů uložených na disku.
- **Vlákna pro uživatelské rozhraní** (*User Interface Threads*) - tato vlákna poskytují přístupové rozhraní mezi Time Machine a uživateli, nebo vzdálenými aplikacemi jako jsou různé IDS systémy.

4.3 Využití Time Machine

Největší využití má tato technika ve spojení s dalšími systémy, pro které má smysl. Tato technika je dobře využitelná detekčními systémy popsanými v kapitole 4.2, jak už jsem uváděl výše, které dokážou tuto možnost retrospektivní analýzy využít v plné míře, při detekci různých útoků, které detekují v reálném čase. Tímto způsobem dokážou přesněji určit, jedná-li se opravdu o útok nebo ne.



Obrázek 4.2: IDS využívající Time Machine [7]

Kapitola 5

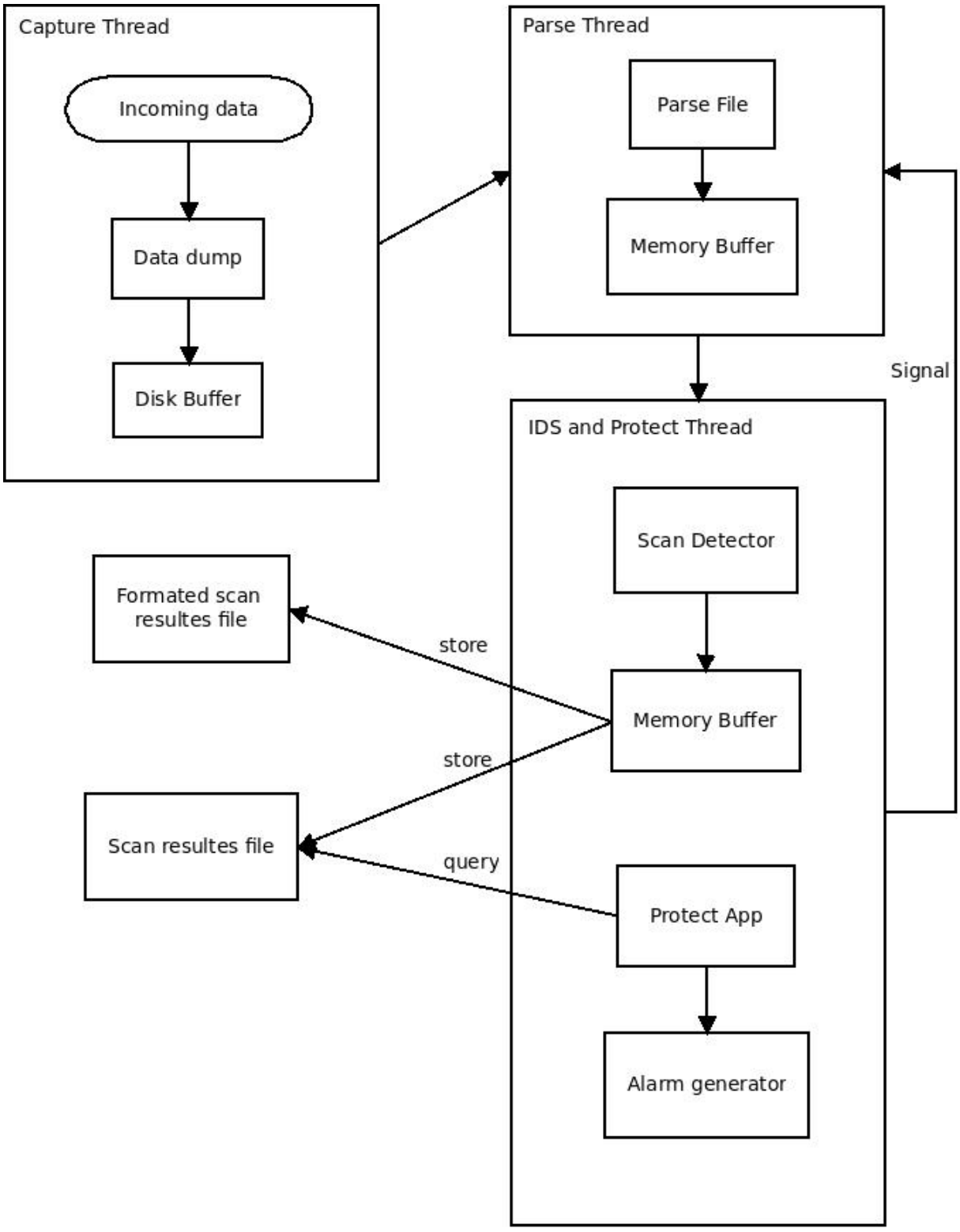
Návrh a implementace aplikace

V praktické částí bakalářské práce jsem se rozhodl navrhnout a implementovat aplikaci, která bude detekovat pokusy o skenování pomocí skenovacích technik popsaných v kapitole 2.5.1. Dále bude mít tato aplikace za úkol chránit konkrétní porty, které budou zadány do seznamu. Tato aplikace bude sloužit jako nástroj pro administrátory sítí, které využívají možnosti NetFlow a bude nástrojem pro zpřesnění detekce o neoprávněný přístup ke konkrétním portům (službám) běžících na počítačích v dané síti. Tato aplikace bude rovněž využívat zjednodušený systém retrospekce, viz. kapitola 4.1. Hlavní myšlenkou bude detekce skenování a detekce připojování se na chráněné porty, a následné zjišťování, zda-li počítač snažící se připojit na chráněný port dříve neskenoval tento počítač. Navržený program bude fungovat jako konzolová aplikace, kde se všechna varování budou zobrazovat na standardní výstup.

5.1 Návrh aplikace

Aplikace se bude skládat ze tří vláken. Dvě z těchto vláken budou mezi sebou komunikovat pomocí signálu, a jedno vlákno bude fungovat zcela nezávisle na zbylých vláknech, viz Obrázek 5.1.

- **Vlákno pro odchytávání a konvertování příchozích dat** - toto vlákno bude mít za úkol zjišťovat, zda přišla nějaká nová data ze sondy nfcapd a následně tato data pomocí nástroje nfdump překonvertovat do textového formátu.
- **Vlákno pro parsování příchozích dat** - toto vlákno rozparsuje příchozí data, která vrátil nfdump a uložil je do textového souboru. Tato rozparsovaná data jsou uložena do dočasné paměti a poskytnuta následujícímu vláknu. Vlákno následně čeká na signál, aby mohlo parsovat další data.
- **Vlákno pro detekci a ochranu** - toto vlákno se skládá z několika částí. Vlákno nejprve čeká na signál, kterým mu signalizuje předchozí vlákno, že má připravená data. Příchozí data jsou nejdříve překopírovány a je zaslán signál předchozímu vláknu, že může pokračovat v načítání dalších dat. Data, která byla získána jsou následně analýzována a je spuštěna detekce skenování. Po této fázi je spuštěna ochrana služeb, kde je prováděna retrospektivní kontrola s detekovanými skeny. Pokud je zjištěno, že došlo k pokusu o připojení se na chráněný port z počítače, který v minulosti skenoval, dojte tak k vyvolání alarmu, který je vypsán na standardní výstup aplikace.



Obrázek 5.1: Návrh aplikace dnFlow

5.2 Pravidla pro detekci

Zde bych rád uvedl všechna pravidla použitá pro detekci skenování pomocí síťových toků.

5.2.1 Pravidla obecně

Syntaxe pravidel použitých v textu. Je popsána Backus-Naurovou formou.

```
<rule> ::= proto <protocol> |
           proto <protocol> and not flags <flags> and flags <flags> and packets
<comp-opr> <packets> |
           proto <protocol> and not flags <not flags> |
           proto <protocol> and packets <comp-opr> <packets> |
           proto <protocol> and imcp-type <icmp-type> and icmp-code <icmp-code>
<protocol> ::= TCP | UDP | ICMP
<comp-opr> ::= < | > | = | >= | <= | == | !=
<flags> ::= <letter-flg> <flags> | <letter-flg>
<letter-flg> ::= A | S | R | U | P | F
<packets> ::= <number> <packets> | <number> <icmp-type> ::= <number>
<icmp-code> ::= <number>
<number> ::= 0 | ... | 9
```

5.2.2 Jednotlivá pravidla pro detekci

Pravidla jsou navržena pro jednotlivé techniky skenování, které jsou nejčastěji využívány pro zjištěni dostupnosti portů (služeb) běžících na konkrétním počítači.

TCP SYN a connect skenování

Pravidla pro detekci jsou pro obě techniky stejná, protože nelze rozlišit tyto dvě techniky v NetFlow datech.

V prvním kroku potřebujeme nalézt množinu toků, která může obsahovat potencionální skeny. Tuto množinu toků získáme tak, že na vstupní data použijeme filtr: `proto TCP and not flags FPU and flags S and packets < 4`.

Následně v takto získané množině toků nalezneme podmnožiny obsahující stejnou zdrojovou a cílovou IP adresu. V takto nalezené podmožině nyní zjistíme počet toků a počet unikátních portů. Pak si rovněž zjistíme jaký byl maximální počet toků dosažený za vteřinu.

Čím větší jsou tyto získané hodnoty, tím vyšší je pravděpodobnost, že se jedná o skenování. Důležité je rovněž určit si práh, který nám bude označovat sken. Určení tohoto prahu není jednoznačné. Pokud tento práh bude vysoký, tak počet nalezených skenů bude menší, ale rovněž počet falešných alarmů bude menší. Pokud práh bude nízký, tak je větší šance na odhalení skenu, ale bude rovněž docházek k více falešným alarmům. Pokud chceme snížit počet falešných alarmů, máme možnost využít toky mezi účastníky tohoto potencionálního skenu v opačném směru. Na to použijeme filtr: `proto TCP and not flags SFPU and flags AR` a rovněž filtr: `proto TCP and not flags RFPU and flags AS`. Tyto toku jsou odpověďmi skenovaného počítače. Pokud je nastaven příznak AR, tak jde o odpověď z uzavřeného portu. Pokud jsou nastaveny příznaky AS, pak dostáváme odpověď, že port je otevřen.

Null skenování

Tato skenovací technika využívá nestandardního nastavení příznaků v TCP komunikaci, proto je detekce tohoto skenu relativně jednoduchá.

Abychom nalezli takovýto sken, stačí nám vyhledat toky, které nemají v TCP komunikaci nastaven žáden příznak. Proto využijeme tento filtr: `proto TCP and not flags ASRUPF`

Takto nalezené toky můžeme okamžitě prohlásit za skeny, protože takovéto toky se v normální TCP komunikaci nevyskytují.

FIN skenování

Nalezení takto provedeného skenování je v TCP komunikaci jednoduché. Jako u předchozí techniky, využívá nestandardního nastavení příznaku, kde v paketu je nastaven pouze příznak FIN. To znamená, že v TCP komunikaci není typické, aby celý tok měl nastaven pouze příznak FIN.

Abychom nalezli takový sken, využijeme filtru, který nám nalezne pouze toky s nastaveným příznakem FIN: `proto TCP and not flags ASRPU and flags F and packets < 2`.

Takto nalezené toky jsme schopni prohlásit za skeny bez další analýzy.

Xmass skenování

Tato technika, jako dvě předchozí, také využívá nestandardního nastavení příznaků v paketu, a proto je detekce tohoto skenování jednoduchá.

Abychom byli schopni skenování pomocí této techniky detekovat, využijeme filtru, který nám nalezne toky obsahující pouze jeden paket a mají nastaveny příznaky Push, Urgent a FIN. Filtr bude definován: `proto TCP and not flags ASR and flags FPU and packets < 2`.

A jako u předchozích dvou technik, můžeme takto nalezené toky prohlásit za skeny.

UDP skenování

V prvním kroku získáme množinu toků, která bude odpovídat zadanému filtru: `proto UDP and packets < 2`.

Následně v takto získané množině toků nalezneme podmožiny obsahující stejnou zdrojovou a cílovou IP adresu. V takto nalezené podmnožině nyní zjistíme počet toků a počet unikátních portů. Pak si rovněž zjistíme, jaký byl maximální počet toků dosažený za vteřinu.

Čím větší jsou tyto získané hodnoty, tím vyšší je pravděpodobnost, že se jedná o skenování. Důležité je, že u normálního provozu se bude maximální počet toků pohybovat v nízkých hodnotách narozdíl od skenování. Jako u TCP SYN skenování jde o to určit práh, který bude označovat co ještě je sken a co už není. Pokud bude tento práh příliš vysoký, tak počet nalezených skenů bude menší, ale rovněž počet falešných alarmů bude menší. Pokud zas bude práh nastaven příliš nízko, tak bude počet nalezených skenů větší, ale rovněž počet falešných alarmů se zvýší. Proto máme rovněž možnost kontrolovat toky mezi účastníky potencionálního skenu v opačném směru. Na to použijeme filtry: `proto ICMP and ICMP-TYPE 3 and ICMP-CODE 3` a `proto UDP`. Tyto toky jsou odpovědi skenovaného počítače. Pokud je počet ICMP toků vysoký, pak se jedná o sken.

Horizontální TCP SYN skenování

Abychom nalezli skeny, které využívají techniku horizontálního skenování, použijeme filtr: `proto TCP and not flags FPU and flags S and packets < 3`.

Následně v takto získané množině toků nalezneme toky, které obsahují pouze stejnou zdrojovou IP adresu. Horizontální sken musí obsahovat minimálně dvě cílové IP adresy. Pokud tato podmínka není splněna, zahodíme takovéto podmnožiny. Pak určíme počet toků, počet unikátních IP adres a rovněž počet unikátních portů. Také si určíme, jaký byl maximální počet toků dosažený za vteřinu.

Čím větší jsou tyto získané hodnoty, tím vyšší je pravděpodobnost, že se jedná o skenování. Jako u předchozích technik (TCP SYN skenování a UDP skenování) jde o určení prahu, který nám bude označovat, co ještě je sken a co už není. Důsledky nastavení prahu jsou popsány u výše vyjmenovaných skenovacích technik.

Detekce konkrétního portu

V navržené aplikaci rovněž používám filtr, kterým získávám podmnožiny toků, kde se nacházejí pouze toky směřující na konkrétní port.

5.3 Implementace

Co se týče implementace, tak navržená aplikace je vytvořena v jazyce C++. Je rozdělená do několika hlavních tříd. Třída `NetflowDataDump` vytvoří vlákno, které kontroluje nově příchozí data z nfcapd a následně je pomocí nfdump převede do textového souboru. Tato třída rovněž dědí od třídy `NetflowData`, která obsahuje pomocné metody. Další třídou je `NetflowDataParser`, která vytvoří další vlákno pro zpracování dat převedených předchozím vláknem. Toto vlákno rozparsuje data a uloží je do seznamu, které následně zpracovává vlákno vytvořené ve třídě `NetFlowIDSScan`. V tomto vlákně probíhá nastavení pravidel pro detekci skenování a rovněž somotná detekce. Pokud je nějaké skenování detekováno, informace o skenování jsou uložena do souboru, aby byly dostupné pro zpětnou analýzu. Následně se zde volají metody z třídy `NetflowProtectApp`, které detekují neoprávněný přístup na konkrétní port (službu), který je chráněn. Pokud je detekován přístup na chráněny port, je použitá retrospektivní analýza s detekovanými skeny. Pokud dojde ke shodě, je vyvolán alarm, který se vytiskne na standardní výstup a zároveň se pomocí `mailx` odešle administrátorovi na email. Všechny výše popsané třídy jsou volány z hlavní funkce `main`, která tvoří hlavní smyčku programu. Ve funkci `main` se rovněž volá funkce `parseArguments` pro parsování vstupních parametrů programu, a také se volá funkce `getSettings`, která načte ze souboru nastavení programu. Rovněž je zde volána funkce `checkSystemTool` pro kontrolu systémových nástroju, které jsou nezbytné pro běh aplikace.

Kapitola 6

Výsledky experimentů

V rámci této kapitoly jsem vytvořil několik experimentů, kterými jsem otestoval vytvořenou aplikaci. Použil jsem dvě testovací sady dat. Jednu sadu testovacích dat jsem si sám vytvořil ze síťového provozu směřující přes moje síťové rozhraní. Tato data jsem získával s využitím nástrojů pro práci s NetFlow protokolem. Testovací útoky jsem prováděl ze svého počítače a díky tomu jsem docílil získání NetFlow dat obsahující podezřelá data. Druhou testovací sadu jsem získal ze stránek projektu WIDE Project [8], kde jsem použil denní úsek od 12:00 do 19:00 z trans-pacifické linky (18Mbps CAR on 100Mbps link).

6.1 Získání dat

Nyní uvedu a stručně popíšu nástroje, které jsem použil pro získání NetFlow dat.

- `fprobe` - je sowtwarová sonda, která snímala síťový provoz na mém síťovém rozhraní a exportovala získaná data do kolektoru
- `nfcapd` - tento nástroj jsem použil jako kolektor, který četl data exportovaná nástrojem `fprobe` a v pětiminutových úsecích je ukládal do souboru
- `softflowd` - je softwarová sonda, která dokázala zpracovat předhozený TCP dump dat, která jsem získal ze stránek WIDE Project [8]. Ta, jako sonda fprobe, následně exportovala data do kolektoru.

6.2 Testovací útoky

Aby získaná data obsahovala útoky, které bych mohl detekovat, provedl jsem ze svého počítače několik testovacích útoků na okolní počítače. Pro ukázku funkčnosti vytvořené aplikace jsem tedy prováděl útoky, které tato aplikace byla schopna detekovat. Hlavním útokem bylo již výše uvedené skenování popsané v kapitole 2.5. Následujícím útokem pro testovací účely byla snaha o prolomení hesla SSH účtu pomocí slovníkového útoku.

6.2.1 Útok skenování

Abych mohl provést útok skenování, tak jsem nejdříve vyhledal programy, které jsou tohoto útoku schopny. Vybral jsem dva nástroje, které jsem využil a to zejména Nmap, viz kapitola 2.5.2, a Nessus, viz kapitola 2.5.2. Díky těmto nástrojům jsem získal data obsahující útok skenování, který by měla vytvořená aplikace detekovat.

6.2.2 Slovníkový útok

Po útoku skenování jsem využil nástroj Hydra, který je schopen provést slovníkové útoky na různé protokoly. Já jsem tento nástroj použil pro pokus o prolomení hesla SSH účtu. V průběhu testu jsem využil slovník s celkem 70,000 slovy. Díky tomu se v mých získaných datech objevilo velké množství toků, které by měla vytvořená aplikace detekovat.

6.3 Výsledky

Zde uvedu výsledky, které jsem získal z obou sad testovacích dat.

6.3.1 Vlastní testovací data

Získaná data se dá považovat za data pocházející z malé sítě, která neobsahují velký počet toků.

V textu jsou použity tyto zkratky:

FPS - počet toků za vteřinu
UP - počet nalezených unikátních portů
UIP - počet nalezených unikátních IP adres
Regulérní přihlášení - připojení se na port, který má aplikace chránit. Nejedná se však o útok.

Test č.1	FPS	UP	UIP
Vertikální	250	6000	-
Horizontální	150	-	200

Test č.2	FPS	UP	UIP
Vertikální	260	7000	-
Horizontální	200	-	250

Tabulka 6.1: Nastavení aplikace

Test č.1		Útoky	Detekce
Skeny	Vertikální	4	5
	Horizontální	1	1
SSH útok		4	3
DoS útok		0	2
Regulérní přihlášení		6	6

Test č.2		Útoky	Detekce
Skeny	Vertikální	4	4
	Horizontální	1	0
SSH útok		4	3
DoS útok		0	2
Regulérní přihlášení		6	5

Tabulka 6.2: Výsledky detekce síťových anomálií

Aplikace byla spuštěna se všemi parametry detekce skenování. To znamená, že aplikace detekovala všechny techniky skenování, pro které byla navržena. Tyto techniky jsou popsány v kapitole 5.2. Dále byla spuštěna s parametrem obsahujícím porty, které mají být touto aplikací chráněny. Tyto porty byly:

- 22 - standardní SSH port, pod kterým běží SSH servery na školních počítačích eva a merlin, tyto stanice jsem nejdříve oskenoval a následně jsem se regulérně přihlásil na jejich SSH server

- `2222` - rovněž SSH port, pod kterým běží SSH server, který mi sloužil pro testování slovníkových útoků

Z výsledku je patrné, že aplikace byla schopna detekovat téměř všechna skenování a všechny útoky, které jsem v testovacích datech provedl. To, že aplikace detekovala útok jako DoS, bylo způsobeno tím, že slovníkový útok provedený na SSH byl na nestandardní port 2222, proto ho aplikace zařadila do DoS útoků. Následkem toho, že bylo detekováno více útoků a skenování než bylo provedeno, je způsobeno tím, že při sběru dat v rozmezí pěti minut, byla data odchycená na konci prvního pětiminutového souboru a na začátku následujícího pětiminutového souboru detekována jako útok nebo skenování v obou souborech. Kolonka regulérní přihlášení říká, že aplikace detekovala připojení IP adresy, která skenovala, na chráněný port. Toto však není považováno za útok. Rovněž můžeme vidět, že různá nastavení aplikace mohou mít vliv na detekci anomálií na síti.

6.3.2 Testovací data projektu WIDE Project

Získané TCP dumpy dat z tohoto projektu byly z denního úseku od 12:00 do 19:00. Pomocí sondy `softflowd` a kolektoru `nfcapd` jsem tyto TCP dumpy převedl do NetFlow protokolu. Kolektor, který ukládal data po pětiminutových úsecích obsahoval cca hodinový úsek TCP dumpu. Po použití nástroje `nfdump` měly tyto pětiminutové soubory:

- velikost cca 650 - 750 Mb
- obsahovaly cca 6 000 000 řádků.

Zpracování:

Zpracování dat	
Doba zpracování jednoho souboru	cca 1h15m
Vytížení CPU	100%
Vytížení RAM	170Mb

Tabulka 6.3: Vytížení

Zde je vidět, že aplikace byla schopna zpracovávat hodinový časový úsek téměř v reálném čase za hodinu a patnáct minut.

Nastavení a výsledky testu:

	FPS	UP	UIP
Vertikální	275	8000	-
Horizontální	200	-	250

Tabulka 6.4: Nastavení aplikace

		Detekce
Skeny	Vertikální	672
	Horizontální	89
SSH útok		0
DoS útok		0
Regulérní přihlášení		1

Tabulka 6.5: Výsledky detekce síťových anomálií

Zde je ukázka nalezených skenů:

Horizontální Skenování
Zdrojová adresa: 203.103.35.130
Čas: 2010-05-22 05:28:36

Fin, Null nebo Xmass Skenování
Zdrojová adresa: 27.103.151.213
Zdrojová adresa: 17.44.11.88
Čas: 2010-05-22 05:29:00

Horizontální Skenování
Zdrojová adresa:24.141.58.132
Čas: 2010-05-22 05:35:00

Z těchto výsledků detekce síťových anomálií můžeme vyvodit, že na takto velké lince, ze které pocházejí data, nemá smysl detekovat vertikální skenování. Většina vertikálního skenování, která byla detekováná byla provedena metodou FIN, NULL nebo Xmass. Smysl má pouze detekce horizontálního skenování. Dále vidíme, že žáden, který skenoval se nepokoušel o útok na SSH ani jiný chráněný port. Byl pouze detekován pokus o připojení se na chráněný port, ten ale nebyl detekován jako útok, nýbrž jako regulérní přihlášení.

6.3.3 Shrnutí výsledků

Aplikace, kterou jsem navrhl a vytvořil, byla schopna v testovacích datech, které jsem sám vytvořil detekovat s vhodným nastavením téměř všechny síťové anomálie, které jsem provedl. V datech, které jsem získal z projektu WIDE project, aplikace detekovala hodně pokusů o skenování, a rovněž detekovala přístup na chráněný port. Tato anomálie, ale nebyla detekována jako útok. Aplikace nedetekovala v těchto datech žáden pokus o útok. Bohužel jsem nebyl schopen zjistit, zda-li v těchto datech nějaké útoky opravdu byly nebo ne.

Kapitola 7

Závěr

Cílem této bakalářké práce, bylo navrhnout aplikaci, která bude využívat data získána pomocí protokolu NetFlow a pomocí nich bude v reálném čase detekovat některé ze síťových útoků. Aplikace tedy byla navržena na detekci skenování sítí s rozšířením na ochranu konkrétních portů (služeb) v dané síti. Aplikace využívá retrospektivní analýzy s detekovanými skeny.

Ze začátku své bakalařské práce jsem se zaměřil na problematiku monitorování sítě s využitím protokolu NetFlow vyvinutým firmou Cisco Systems. Tento protokol jsem popsal jak z pohledu jeho architektury, tak rovněž z hlediska jeho funkčnosti, a taky z hlediska využití tohoto protokolu. Společně s touto technologii jsem rovněž představil vybrané nástroje pro práci s daty získanými pomocí protokolu NetFlow.

V další části mé práce jsem se věnoval analýze síťových útoků, kde jsem popsal principy fungování několika známých útoků, se kterými se můžeme často setkávat na síti. V této části jsem se především zaměřil na popis fungování skenování sítě, kterým jsem se dále zabýval.

Následně jsem se zabýval konkrétními možnostmi detekce útoků na síti s využitím detekčních mechanismů. Zde jsem popsal rozdíly mezi Intrussion Detection System (IDS) a Intrussion Prevention System (IPS). V této kapitole jsem rovněž rozebral možnosti detekce skenování sítě s pomocí NetFlow protokolu. Rovněž jsem popsal, jak se tato anomálie v NetFlow datech projevuje.

Následující kapitola se zabývá mechanismem Time Machine, který je popsán z hlediska jeho architektury a rovněž z hlediska jeho využití různými detekčními mechanismy.

Dále jsem popsal návrh a implementaci aplikace, kterou jsem vytvořil v rámci praktické části bakalářské práce. Zde jsem se hlavně zaměřil na popis pravidel pro detekci skenování. Jsou zde popsány pravidla pro vybrané techniky skenování na síti.

V poslední části své bakalářské práce jsem vytvořil několik experimentů, kterými jsem testoval navrženou aplikaci. Při experimentech jsem použil data, která jsem si sám vytvořil použitím softwarové sondy `fprobe` a programu `nfcapd`. Experimentální data pocházejí z vnitřní sítě VUT. Na závěr jsem zhodnotil dosažené výsledky, které byly v použitých datech pozitivní. Aplikace dokázala detekovat skenování, která se v datech vyskytovala a také detekovala snahu o neoprávněný přístup k portům (službám), které měla chránit. Co se týče rozšíření programu, mohlo by se dále navrhnout a implementovat moduly pro detekci různých DoS útoků.

Literatura

[1] CISCO Systems: Introduction to Cisco IOS NetFlow - A Technical Overview. online, [cit. 2010-03-20].
URL http://www.ciscostadium.org/en/US/prod/collateral/iosswrel/ps6537/\ps6555/ps6601/prod_white_paper0900aecd80406232.pdf

[2] CISCO Systems: NetFlow Services Solutions Guide. online, [cit. 2010-03-20].
URL
http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.html

[3] Grégr, M.: *Detekce a izolace útočníků pomocí záznamů NetFlow*. Diplomová práce, FIT VUT v Brně, 2009.

[4] Ierace, N.; Urrutia, C.; Bassett, R.: Intrusion prevention systems. *Ubiquity*, ročník 2005, č. June: s. 2–2.
URL http://www.acm.org/ubiquity/views/pf/v6i19_ierace.pdf

[5] Kompella, R. R.; Singh, S.; Varghese, G.: On scalable attack detection in the network. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA: ACM, 2004, ISBN 1-58113-821-0, s. 187–200.
URL http://www.imconf.net/imc-2004/papers/p187-kompella.pdf

[6] Lee, C. B.; Roedel, C.; Silenok, E.: Detection and Characterization of Port Scan Attacks. online, [cit. 2010-04-18].
URL http://cseweb.ucsd.edu/~clbailey/PortScans.pdf

[7] Maier, G.; Sommer, R.; Dreger, H.; aj.: Enriching network security analysis with time travel. In *SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, New York, NY, USA: ACM, 2008, ISBN 978-1-60558-175-0, s. 183–194.
URL http://www.icir.org/vern/papers/time-machine-sigcomm08.pdf

[8] MAWI Working Group: WIDE Project. online, [cit. 2010-04-26].
URL http://tracer.csl.sony.co.jp/mawi/

[9] Mirkovic, J.; Dietrich, S.; Dittrich, D.; aj.: *Internet Denial of Service: Attack and Defense Mechanisms*. Prentice Hall PTR, Prosinec 2004, iSBN 0-13-147573-8.

[10] Mirkovic, J.; Reiher, P.: A taxonomy of DDoS attack and DDoS defense mechanisms. *SIGCOMM Comput. Commun. Rev.*, ročník 34, č. 2, 2004: s. 39–53, ISSN 0146-4833.
URL http://doi.acm.org/10.1145/997150.997156

[11] NESSUS: The Network Vulnerability Scanner. online, [cit. 2010-04-18].
URL http://www.nessus.org/nessus/

[12] NFDUMP: Nfdump tools. online, [cit. 2010-04-18].
URL http://nfdump.sourceforge.net/

[13] NMAP: Nmap Security Scanner. online, [cit. 2010-04-18].
URL http://nmap.org/

[14] Peng, T.; Leckie, C.; Ramamohanarao, K.: Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Comput. Surv.*, ročník 39, č. 1, 2007: str. 3, ISSN 0360-0300.
URL http://doi.acm.org/10.1145/1216370.1216373

[15] Rehák, M.; Pěchounek, D.; Čeleda, P.; aj.: Camnep: An intrusion detection system for highspeed networks. online, 2008, [cit. 2010-04-18].
URL http://www.nii.ac.jp/pi/n5/5_65.pdf

[16] Richard, M.: Intrusion Detection FAQ: Are there limitations of Intrusion Signatures? online, Duben 2001, [cit. 2010-04-18].
URL http://www.sans.org/security-resources/idfaq/limitations.php

[17] Scarfone, K.; Mell, P.: Guide to Intrusion Detection and Prevention Systems (IDPS). online, Únor 2007, [cit. 2010-04-18].
URL http://csrc.ncsl.nist.gov/publications/nistpubs/800-94/SP800-94.pdf

[18] Sommer, P.: Intrusion detection systems as evidence. *Comput. Netw.*, ročník 31, č. 23-24, 1999: s. 2477–2487, ISSN 1389-1286.
URL
http://raid-symposium.org/raid98/Prog_RAID98/Full_Papers/Sommer_text.pdf

Dodatek A

Obsah CD

Obsahem CD jsou zdrojové kódy vytvořené aplikace, včetně přiloženého Makefilu. Dále jsou na CD ukázková testovací data, README a také dokumentace k projektu. Rovněž CD obsahuje pdf soubor této bakalářské práce.

Dodatek B

Požadavky na systém

Vytvořená aplikace byla testována v prostředí Linux distribuce Ubuntu 9.10. Všechny zdrojové kódy se nacházejí na CD včetně Makefilu, který využijte pro překlad aplikace. Pro kompilaci je nutné mít nainstalovaný překladač g++. Pro funkčnost je vyžadován rovněž nástroj `nfdump` verze 1.5.7 pro konverzi NetFlow dat z kolektoru `nfcapd` do textového souboru. Rovněž pro využití e-mailových upozornění je nutná funkčnost nástroje `mailx`. Tento nástroj není podmínkou funčnosti aplikace.

Dodatek C

Manual

Vytvořená aplikace je bez grafického rozhraní a ovládá se pomocí příkazového řádku a konfiguračního souboru.

Aplikace přijímá tyto parametry:

-cF <nazev souboru>	název aktuálního souboru, který je bez předpony nfcapd
-v	parametr, který aktivuje detekci vertikálního skenování (defaultně nastaven)
-h	parametr, který aktivuje detekci horizontální skenování
-sS	parametr, který aktivuje detekci skenování pomocí metody SYN Scan (defaultně nastaven)
-uS	parametr, který aktivuje detekci skenování pomocí metody UDP Scan
-fnxS	parametr, který aktivuje detekci skenování pomocí metody FIN, NULL nebo Xmass Scan

Dále je aplikace ovládána pomocí konfiguračního souboru `settings.txt`:

parseLine=<počet řádků>	počet řádků, které se zpracují v jednom cyklu
inputPath=<cesta>	cesta ke vstupním netflow datům
outputPath=<cesta>	výstupní cesta, kde se budou ukládat data překonvertované pomocí nástroje `nfdump`
mailing=<bool>	povolení či zakázání posílání e-mailů s varováním (true/false)
mailAddr=<email>	e-mailová adresa administrátora, na kterou budou zasílána varování
checkSystemTools=<bool>	povolení či zakázání kontroly potřebných systémových nástrojů (true/false)
printScans=<bool>	povolení či zakázání vypisování detekovaných skenů
Rozšířené nastavení	
tresholdVerticalFPS=<číselná hodnota>	práh pro detekci vertikálního skenování (FPS - počet toků za sekundu)
tresholdVerticalUP=<číselná hodnota>	práh pro detekci vertikálního skenování (UP - počet unikátních portů)
tresholdHorizontalFPS=<číselná hodnota>	práh pro detekci horizontálního skenování (FPS - počet toků za sekundu)
tresholdHorizontalUIP=<číselná hodnota>	práh pro detekci horizontálního skenování (UIP - počet unikátních IP adres)