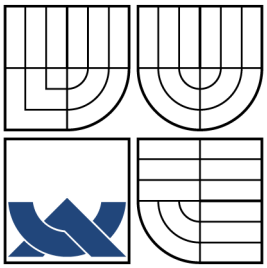
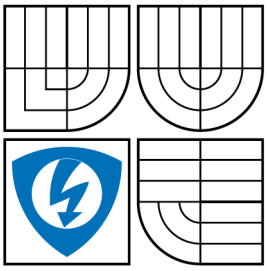


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

POKROČILÉ METODY FILTROVÁNÍ SÍŤOVÉHO PROVOZU V LINUXU

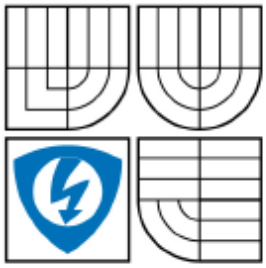
ADVANCED METHODS OF FILTERING NETWORK TRAFFIC IN THE LINUX SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — BC.DAVID PEŠA
AUTHOR

VEDOUCÍ PRÁCE — ING.JAN KACÁLEK
SUPERVISOR

BRNO 2008



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Peša David Bc.	*ID:*	88956
Ročník:	2	*Akademický rok:*	2007/2008

NÁZEV TÉMATU:

Pokročilé metody filtrování síťového provozu v systému Linux

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte konfiguraci firewallu pro systém GNU/Linux. Řešení bude obsahovat návrh topologie sítě, specifikaci možných bezpečnostních rizik a metody implementace preventivních opatření do firewallu. Dále se zaměřte na způsoby integrace systémů IPS a IDS do prostředí GNU/Linux, proveďte obsahově založené filtrování na vrstvě layer 7. Navrhněte a implementujte jednu z metod zabezpečení datových toků (VPN). Proveďte audit síťových logů.

DOPORUČENÁ LITERATURA:

[1] Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman, Oreilly - Building Internet Firewalls, June 2000, ISBN: 1-56592-871-7, 890 pages
[2] Olaf Kirch & Terry Dawson, Linux Network Administrator's Guide, June 2000, ISBN: 1-56592-400-2

Termín zadání:	11.2.2008	*Termín odevzdání:*	28.5.2008

Vedoucí práce: Ing. Jan Kacálek

prof. Ing. Kamil Vrba, CSc.
předseda oborové rady

UPOZORNĚNÍ:

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. **Pan/paní**

   Jméno a příjmení: Bc. David Peša
   Bytem: Ukrajinská 546/21, 62500, Brno - Bohunice
   Narozen/a (datum a místo): 20.12.1983, Brno

(dále jen "autor")

a

2. **Vysoké učení technické v Brně**

   Fakulta elektrotechniky a komunikačních technologií
   se sídlem Údolní 244/53, 60200 Brno 2
   jejímž jménem jedná na základě písemného pověření děkanem fakulty:
   prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ☐ disertační práce
   - ☑ diplomová práce
   - ☐ bakalářská práce

   jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Pokročilé metody filtrování síťového provozu v systému Linux
Vedoucí/školitel VŠKP: Ing. Jan Kacálek
Ústav: Ústav telekomunikací
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

- ☑ tištěné formě - počet exemplářů 1
- ☑ elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ☑ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....	.....
Nabyvatel	Autor

ABSTRAKT

Tato diplomová práce se zabývá technikami návrhu a vytváření samotného paketově filtračního firewallu v systému Linux, především pro malé sítě, které nenabízí mnoho služeb internetovým uživatelům. Dále se pojednává o zmírňování účinků nejběžnějších typů útoků za použití iptables. Tato práce pojednává o metodách návrhu, implementace, provozu a údržby firewallu. Implementovány jsou techniky pro průběžné monitorování útoků. Také je zmíněn historický, architektonický a technický přehled firewallů a bezpečnostních útoků.

KLÍČOVÁ SLOVA

Firewall, IDS, IPS, útok, linux, iptables, VPN, zabezpečení

ABSTRACT

This master's thesis is meant to provide techniques in designing and building a standalone packet filtering firewall in Linux machines, mainly for small sites who don't give much service to Internet users. It deals with attenuating the effect of the most common types of attacks using iptables. It guides how to design, implement, run, and maintain Firewall. Techniques for continuously monitoring attacks is attempted. It also give a historical, architectural and technical overview of firewalls and security attacks.

KEYWORDS

Firewall, IDS, IPS, attack, linux, iptables, VPN, security

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Pokročilé metody filtrování síťového provozu v linuxu“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

(podpis autora)

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ACL Seznam pro řízení přistupu k síťovým zdrojům

DHCP Protokol pro dynamickou konfiguraci nejen IP adres

DMZ Zóna se speciálními požadavky na bezpečnou komunikaci

DNAT Překlad cílových síťových adres

DoS Typ útoky mající za cíl vyřadit počítačový systém z provozu

FTP Protokol aplikační vrstvy pro přenos souborů mezi počítači

GRE Tunelovací protokol pro zapouzdření různých paketů síťové vrstvy

HIDS IDS systém monitorující vnitřní počítačový systém

IDS Systém pro detekci útoků

IPv4 Protokol IP ve verzi 4, v současnosti hojně používán v IP sítích

IPv6 Protokol IP ve verzi 6, nová generace protokolu IP a nástupce protokolu IPv4

NAT Překlad síťových adres

NIDS IDS systém monitorující podezřelou aktivitu v síťi

OSI Normy pro propojování síťových systémů vydávané organizací ITU-T

PPTP Tunelovací protokol implementovaný ve virtuálních privátních sítích

RFC Standardy a normy pro počítačové systémy vydávané organizací IETF

SNAT Zdrojový překlad síťových adres

SNORT Systém NIDS implementující prvky analýzy datového provozu v reálném čase

SSH Síťový protokol umožňující výměnu dat zabezpečeným kanálem

TCP/IP Sada internetových IP protokolů

TTL Parametr doby životnosti paketů

WWW Označení pro aplikace založené na protokolu HTTP

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

V této diplomové práci bych se rád věnoval metodám filtrování datového provozu a specifikováním bezpečnostních rizik s návrhem protiopatření. Základní principy směrování, činnost protokolů TCP/IP, navazování spojení a funkce firewallu budou základním tématem pro pochopení problematiky. V úvodní části řešení se zaměřím na metody návrhu firewallu, popis nejběžnějších útoků na síťovou infrastrukturu a návrhu preventivních opatření s optimální konfigurací. Předmětem další části této práce bude specifikace vlastností systémů IPS/IDS a jejich implementace na modelovou síť spolu s metodou zabezpečení datových toků. V poslední části této práce se zaměřím na způsoby analýzy síťového provozu. Cílem projektu je navrhnout a implementovat řešení pro zabezpečení menší síťě. Výstupem bude sada skriptů pro automatickou konfiguraci firewallu a bezpečnostních prvků.

Firewall izoluje vnitřní síť od internetu. Prověřuje každý paket, který k němu dorazí z jedné či z druhé strany podle toho, zda přichází z vnější sítě či naopak. Určuje, zda jim má být umožněn průchod či mají být zastaveny. Obecně lze říci, že nástroj typu firewallu slouží především pro zabezpečení vstupního bodu do sítě internetu, nebo jiné veřejné sítě. Firewall je především soubor opatření, která umožňují např. řízení přístupu uživatele z vnější i vnitřní sítě, nastavení přístupových práv, odfiltrování nebezpečných služeb, soustředění bezpečnosti do jednoho komunikačního uzlu, zablokování nežádoucího mapování vnitřní sítě, audit platných i neplatných operací. Firewall je svojí definicí a jedinečným umístěním v topologii informačního systému lehce využitelný k tomu, aby zahrnul další funkční rysy, které od něj v současnosti nejsou ještě zcela samozřejmě vyžadovány. Tyto nadstandardní vlastnosti mají přímo i nepřímo pozvednout úroveň zabezpečení celého systému.

1 FIREWALLY

Firewall je síťové zařízení, které slouží k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti nebo zabezpečení. Zjednodušeně se dá říct, že slouží jako kontrolní bod, který definuje pravidla pro komunikaci mezi sítěmi, které od sebe odděluje. Tato pravidla historicky vždy zahrnovala identifikaci zdroje a cíle dat (zdrojovou a cílovou IP adresu) a zdrojový a cílový port, což je však pro dnešní firewally už poměrně nedostatečné. Modernější firewally se opírají přinejmenším o informace o stavu spojení, znalost kontrolovaných protokolů a případně prvky IDS. Firewally se během svého vývoje řadily zhruba do následujících kategorií:

1. **Paketové filtry**
2. **Aplikační brány**
3. **Stavové paketové filtry**
4. **Stavové paketové filtry kombinované s** IDS

1.1 Paketové filtry

Nejjednodušší a nejstarší forma činnost firewallu spočívá v tom, že pravidla přesně uvádějí, z jaké adresy a portu na jakou adresu a port může být doručen procházející paket, tj. kontrola se provádí na třetí a čtvrté vrstvě modelu síťové komunikace OSI. Výhodou tohoto řešení je vysoká rychlost zpracování, proto se ještě i dnes používají na místech, kde není potřebná přesnost nebo důkladnější analýza procházejících dat, ale spíš jde o vysokorychlostní přenosy velkých množství dat. Nevýhodou je nízká úroveň kontroly procházejících spojení, která zejména u složitějších protokolů (např. FTP, video/audio streaming, apod.) nejen nedostačuje ke kontrole vlastního spojení, ale pro umožnění takového spojení vyžaduje otevřít i porty a směry spojení, které mohou být využity jinými protokoly, než bylo zamýšleno. Mezi typické představitele paketových filtrů patří ACL (Seznam pro řízení přistupu k síťovým zdrojům) ve starších verzích operačního systému IOS na routerech Cisco Systems,u starší varianty firewallu v linuxovém jádře (ipchains).

1.2 Aplikační brány

Jen o málo později, než jednoduché paketové filtry, byly představeny firewally, které na rozdíl od paketových filtrů zcela oddělily sítě, mezi které byly postaveny. Říká se jim většinou Aplikační brány, někdy také Proxy firewally. Veškerá komunikace přes

aplikační bránu probíhá formou dvou spojení – klient (iniciátor spojení) se připojí na aplikační bránu (proxy), ta příchozí spojení zpracuje a na základě požadavku klienta otevře nové spojení k serveru, kde klientem je aplikační brána. Data, která aplikační brána dostane od serveru pak zase v původním spojení předá klientovi. Kontrola se provádí na sedmé vrstvě síťového modelu OSI ,proto se těmto firewallům říká aplikační brány.

Jedním vedlejším efektem použití aplikační brány je, že server nevidí zdrojovou adresu klienta, který je původcem požadavku, ale jako zdroj požadavku je uvedena vnější adresa aplikační brány. Aplikační brány se díky tomu chovají jako nástroje pro překlad adres - NAT. Výhodou tohoto řešení je poměrně vysoké zabezpečení. Nevýhodou je zejména vysoká náročnost na použitý hardware. Aplikační brány jsou schopny zpracovat mnohonásobně nižší množství spojení a rychlosti, než paketové filtry a mají mnohem vyšší latenci. Každý protokol vyžaduje napsání specializované proxy, nebo použití univerzální proxy, která ale není o nic bezpečnější, než využití paketového filtru. Původní aplikační brány navíc vyžadovaly, aby klient uměl s aplikační branou komunikovat a neuměly dost dobře chránit svůj vlastní operační systém. Tyto nedostatky se postupně odstraňovaly, ale po nástupu stavových paketových filtrů se vývoj většiny aplikačních bran postupně zastavil a ty přeživší se dnes používají už jen ve velmi specializovaných nasazeních.

1.3 Stavové paketové filtry

Stavové paketové filtry provádějí kontrolu stejně jako jednoduché paketové filtry, navíc si však ukládají informace o povolených spojeních, které pak mohou využít při rozhodování, zda procházející pakety patří do již povoleného spojení a mohou být propuštěny, nebo zda musí znovu projít rozhodovacím procesem. Tato vlastnost přináší dvě výhody, jednak se tak urychluje zpracování paketů již povolených spojení, za druhé lze v pravidlech pro firewall uvádět jen směr navázání spojení a firewall bude samostatně schopen povolit i pakety s odpovědí a u známých protokolů i další spojení, která daný protokol používá.

Například pro FTP tedy stačí nastavit pravidlo, ve kterém povolíme klientu připojení na server pomocí FTP a protože se jedná o známý protokol, firewall sám povolí navázání řídícího spojení z klienta na port 21 serveru, odpovědi z portu 21 serveru na klientem použitý zdrojový port a po příkazu, který vyžaduje přenos dat, povolí navázání datového spojení z portu 20 serveru na klienta na port, který si klient se serverem dohodli v rámci řídícího spojení a pochopitelně i pakety odpovědí z klienta zpět na port 20 serveru. Zásadním vylepšením je i možnost vytváření virtuálního stavu spojení pro bezstavové protokoly UDP a ICMP. K největším výho-

dám stavových paketových filtrů patří jejich vysoká rychlost, poměrně slušná úroveň zabezpečení a ve srovnání s výše zmíněnými aplikačními branami a jednoduchými paketovými filtry řádově mnohonásobně snazší konfiguraci. Zjednodušení konfigurace vede i k nižší pravděpodobnosti chybného nastavení pravidel. Nevýhodou je obecně nižší bezpečnost oproti aplikačním branám.

1.4 Stavové paketové filtry kombinované s IDS

Moderní stavové paketové filtry kromě informací o stavu spojení a schopnosti dynamicky otevírat porty pro různá řídící a datová spojení složitějších známých protokolů implementují technologie Deep Inspection nebo také Application Intelligence. Tyto technologie doplňují firewall o schopnost kontrolovat procházející spojení až na úroveň korektnosti procházejících dat známých protokolů i aplikací. Mohou tak například zakázat průchod http spojení, v němž objeví indikátory, že se nejedná o požadavek na WWW server, ale tunelování jiného protokolu, nebo když data v hlavičce e-mailu nesplňují požadavky RFC apod.

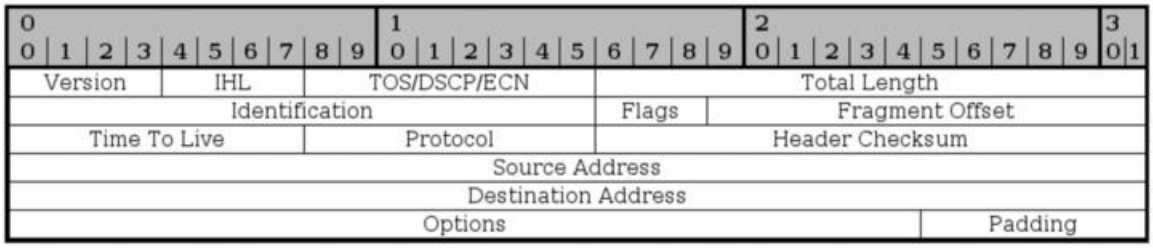
V poslední době se do firewallů integrují tzv. in-line IDS. Tyto systémy pracují podobně jako antivirový software a pomocí databáze signatur a heuristické analýzy jsou schopny odhalit dle vzorků útoky i ve zdánlivě nesouvisejících pokusech o spojení, např. skenování adresního rozsahu, rozsahu portů, známé signatury útoků uvnitř povolených spojení apod. Výhodou těchto systémů je vysoká úroveň bezpečnostní kontroly procházejících protokolů při zachování relativně snadné konfigurace, poměrně vysoká rychlost kontroly ve srovnání s aplikačními branami, nicméně je znát významné zpomalení proti stavovým paketovým filtrům. Nevýhodou je zejména to, že z hlediska bezpečnosti návrhu je základním pravidlem bezpečnosti udržovat bezpečnostní systémy co nejjednodušší a nejmenší. Tyto typy firewallů integrují obrovské množství funkcionality a zvyšují tak pravděpodobnost, že v některé části jejich kódu bude zneužitelná chyba, která povede ke kompromitování celého systému. Podobná funkcionalita je k dispozici ve formě experimentálních modulů také pro iptables v linuxovém jádře.

2 PROTOKOLY

Protokoly jsou základem komunikace v moderních datových sítích postavených na sadě TCP/IP. Z hlediska filtrování datových spojení je třeba si představit protokoly jimiž se tato spojení uskutečňují.

2.1 IP - Internet Protokol

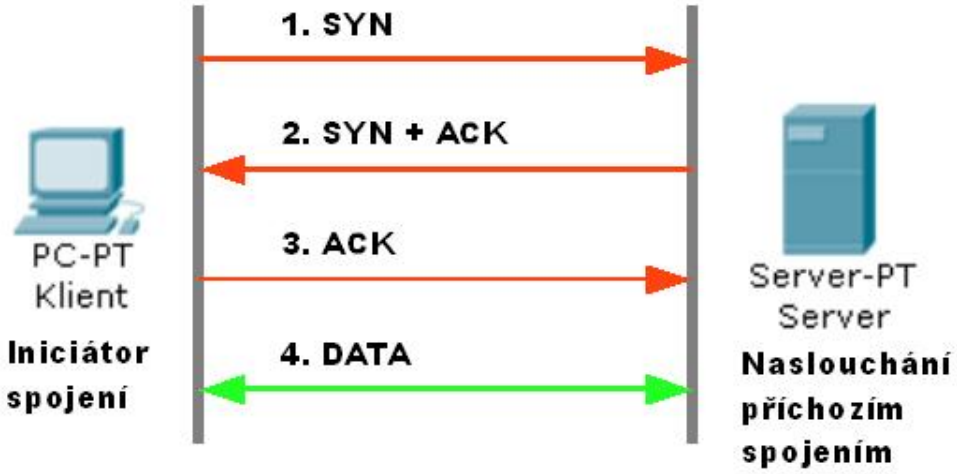
Dnes se nejčastěji používá verze označovaná číslem 4, nazývaná IPv4. IPv6 je navrhovaný a chystaný nástupce IPv4. Každé síťové rozhraní komunikující prostřednictvím IP má přiřazeno jednoznačný identifikátor - IP adresu. V každém datagramu je pak uvedena IP adresa odesilatele i příjemce. Na základě IP adresy příjemce pak každý počítač na trase provádí rozhodnutí, jakým směrem paket odeslat. Data se v IP síti posílají po blocích nazývaných datagramy. Jednotlivé datagramy putují sítí zcela nezávisle, na začátku komunikace není potřeba navazovat spojení, přestože spolu třeba příslušné koncové body nikdy předtím nekomunikovaly. Záhlaví, které musí obsahovat každý IP paket je na obr. 2.1



Obr. 2.1: Hlavička protokolu IP

2.2 TCP - Transmission Control Protocol

U TCP protokolu je nejprve třeba vytvořit spojení. Pro navázání spojení se používá třícestný handshake uvedený na obr 2.2. V průběhu navazování spojení se obě strany dohodnou na sekvenčním čísle. Číslo sekvence a odpovědi jsou 32bitové hodnoty uváděné v TCP hlavičce. Pro navázání spojení se posílá TCP segment, který má nastaveny v TCP hlavičce příznaky, které mohou být: CWR (Congestion Window Reduced), ECE (ECN-Echo), URG (Urgent), ACK (Acknowledgement), PSH (Push), RST (Reset), SYN (Synchronize), FIN. Hlavička protokolu TCP je uvedena na obr. 2.3



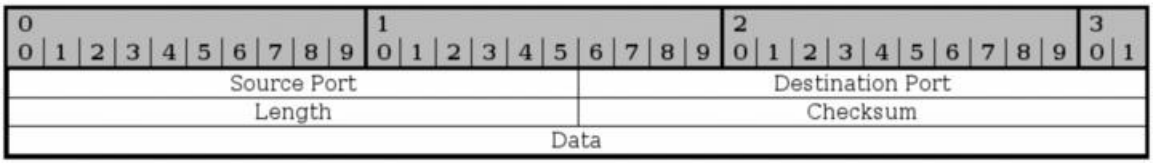
Obr. 2.2: Průběh navazování TCP spojení

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Source Port										Destination Port																					
Sequence Number																															
Acknowledgment Number																															
Data Offset			Reserved				cwr	ece	urg	ack	psh	rst	syn	fin	Window																
Checksum																				Urgent Pointer											
Options																													Padding		
Data																															

Obr. 2.3: Hlavička protokolu TCP

2.3 UDP - User Datagram Protocol

UDP je nespolehlivý bezestavový protokol ze sady protokolů internetu. UDP protokol přenáší datagramy mezi počítači v síti, ale na rozdíl od TCP nezaručuje, že se přenášený paket neztratí, nedojde ke změně pořadí paketů, nebo zda se některý paket nedoručí vícekrát. Díky tomu je UDP pro lehké a časově citlivé účely rychlejší a efektivnější. Jeho bezestavová povaha je také užitečná pro servery, které zodpovídají malé dotazy pro mnoho klientů. V sadě protokolů internetu poskytuje UDP velmi jednoduché rozhraní mezi síťovou vrstvou a aplikační vrstvou. UDP neposkytuje žádné záruky doručení a UDP vrstva odesílatele si u jednou už odeslaných zpráv neudržuje žádný stav. UDP pouze přidává kontrolní součty a schopnost třídit UDP pakety mezi více aplikací běžících na stejném počítači. Hlavička protokolu UDP je uvedena na obr. 2.4



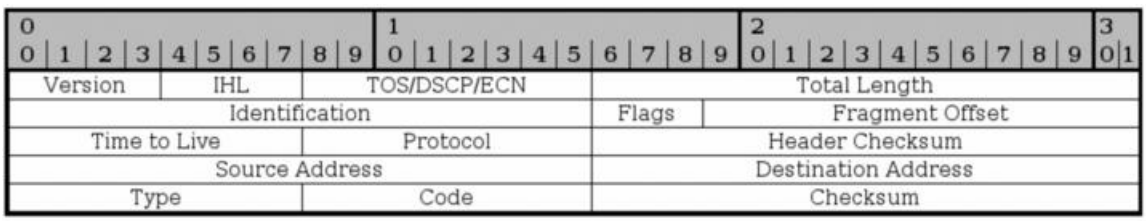
Obr. 2.4: Hlavička protokolu UDP

2.4 ICMP - Internet Control Message Protocol

ICMP protokol je jeden z hlavních protokolů internetu. Používají ho operační systémy počítačů v síti pro odesílání chybových zpráv například pro oznámení, že požadovaná služba není dostupná nebo že potřebný počítač nebo router není dosažitelný. ICMP se svým účelem liší od TCP a UDP protokolů tím, že se obvykle nepoužívá sítovými aplikacemi přímo. Jedinou výjimkou je nástroj ping, který posílá ICMP zprávy "Echo Request" a očekává příjem zprávy "Echo Reply" aby určil, zda je cílový počítač dosažitelný a jak dlouho paketům trvá, než se dostanou k cíli a zpět.

ICMP zprávy se vytváří nad IP vrstvou, obvykle z IP datagramu, který ICMP reakci vyvolal. IP vrstva danou ICMP zprávu zapouzdří s novou IP hlavičkou a obvyklým způsobem vzniklý datagram odešle.

Každý router, který směruje IP datagram, musí v IP hlavičce dekrementovat políčko TTL. Jestliže TTL klesne na hodnotu 0 a datagram není určen koncovému zařízení provádějící dekrementaci pak router přijatý paket zahodí a původnímu odesilateli datagramu pošle ICMP zprávu číslo 11 jak uvádí tabulka 2.2. Ačkoli ICMP zprávy jsou obsažené ve standardních IP datagramech, ICMP zprávy se zpracovávají odlišně od normálního zpracování prokolů nad IP. V mnoha případech je nutné prozkoumat obsah ICMP zprávy a doručit patřičnou chybovou zprávu aplikaci, která vyslala původní IP paket jenž způsobil odeslání ICMP zprávy k původci. Formát protokolu ICMP je uveden na obr. 2.5. Nejdůležitější položky z hlavičky tohoto protokolu jsou pole specifikující typy zpráv viz tab. 2.2 a kódy zpráv uvedené v tab. 2.1.



Obr. 2.5: Hlavička protokolu ICMP

Tab. 2.1: Přehled kódů ICMP zpráv.

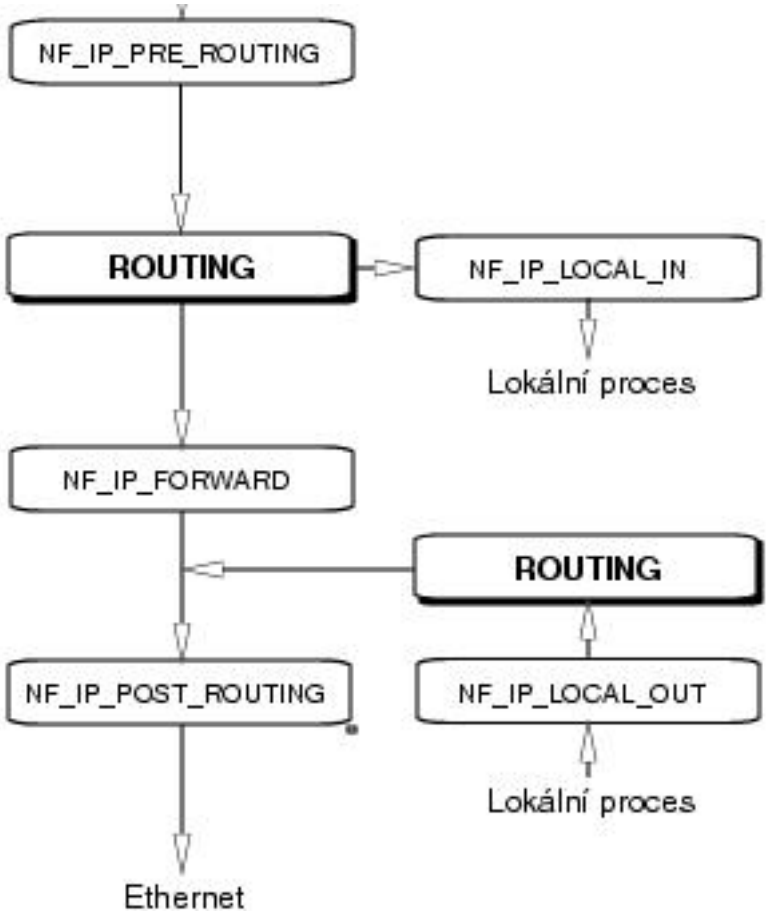
Kód	**Význam kódu**
0	nedosažitelná síť (network unreachable)
1	nedosažitelný uzel (host unreachable)
2	nedosažitelný protokol (protocol unreachable)
3	nedosažitelný port (port unreachable)
4	nedosažitelná síť (network unreachable)
5	nutná fragmentace, ale není povolena
6	neznámá cílová síť (destination network unknown)

Tab. 2.2: Přehled typů ICMP zpráv.

Typ	**Význam typu**
0	odpověď na echo (Echo Reply)
3	adresát nenalezen (Destination Unreachable)
4	žádosti o zpomalení (Source Quench) - tento parametr zajišťuje základní mechanismus pro řízení toku; pokud datagramy přicházejí na směrovač příliš rychle a směrovač je nestíhá zpracovávat, musí je rušit. Při tom posílá zdrojovému uzlu ICMP zprávy Source Quench, jimiž žádá uzel o zpomalení.
5	přesměrování (Redirect) - ICMP redirect je mechanismus, kterým může směrovač upozornit na to, že jiný směrovač v síti umí lepší cestu k cíli; je to zejména v případě, že v síti jsou dva směrovače, přičemž jeden z nich je pro určité uzly odchozí bránou a pakety pro určité sítě přehazuje na ten druhý; potom může upozornit uzel na to, že může pro použít ten druhý směrovač přímo
8	požadavek na echo (Echo Request)
11	překročení času (time exceeded) - generováno směrovačem v případě překročení TTL, generováno uzlem v případě, že se nepodaří defragmentace (některý z fragmentů je ztracen).
12	problém s parametrem (Parameter Problem) - směrovač nebo uzel zjistí problém s některým z parametrů hlavičky; zpráva obsahuje číslo oktetu ve kterém byl problém zjištěn
13	požadavek na timestamp (Timestamp Request)
14	odpověď timestamp (Timestamp Reply)
15	požadavek na informaci o síti (Information Request)
16	odpověď informace (Information Reply)
17	požadavek na masku (Address Mask Request)
18	odpověď masky (Address Mask Reply)

3 NETFILTER

Základem filtrování datových spojení je netfilter framework, který je zpravidla součástí linuxového jádra. Ten definuje na cestě paketu TCP/IP zásobníkem pět bodů, ve kterých se mohou moduly jádra přihlásit k jejich odběru a rozhodnout, zda paketu dají šanci postoupit dál, nebo zda bude zrušen. V každém bodě může být přihlášeno libovolné množství modulů, a pakety, pokud nejsou některým modulem odstraněny pak jsou předány postupně všem. Cestu paketu znázorňuje obr. 3.1



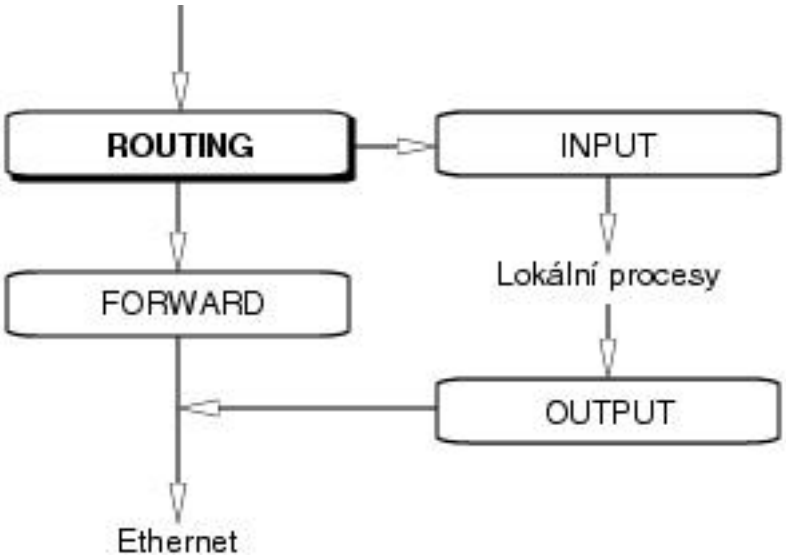
Obr. 3.1: Netfilter framework

Netfilter dále nabízí už jen základní infrastrukturu pro komunikaci kernelu s uživatelským prostředím, kterou lze použít k předávání parametrů do modulů a zpětnému získávání stavových informací.

3.1 Iptables

Vlastní filtr paketů je implementován sadou modulů jádra, které je nutno zavést před vlastním vytvářením filtrovacích pravidel. Rozčlenění na moduly různého druhu

přináší řadu výhod, například lze zavádět pouze ty moduly, které jsou potřebné. Iptables jsou navrženy jako rozšiřitelné právě pomocí modulů jádra. Jako moduly lze přidávat jednak cíle a také filtrovací pravidla. Aby bylo možné spravovat tyto moduly pomocí nástroje iptables, je i tento program navržen jako rozšiřitelný pomocí dynamických knihoven. Každý pokročilejší modul jenž vyžaduje předat nějaké parametry má svůj doplněk v dynamické knihovně pro iptables. Cesta paketu filtrovacím kódem poskytuje větší nezávislost filtrovacích pravidel na rozhraních. Graficky znázorněno na obr. 3.2



Obr. 3.2: Průchod paketu základními řetězci iptables

Je tu jistá analogie s předchozím obrázkem 3.1 kde řetězec INPUT odpovídá bodu NF_IP_LOCAL_IN, řetězec FORWARD bodu NF_IP_FORWARD a řetězec OUTPUT bodu NF_IP_LOCAL_OUT. Paket procházející řetězcem FORWARD, ve kterém se v běžných případech odehrává většina filtrování ví jak o vstupním, tak i o výstupním rozhraní jelikož již bylo rozhodnuto o jeho směrování. Kromě řetezců INPUT, OUTPUT a FORWARD v tabulce filter existují ještě další, například v tabulce nat PREROUTING, POSTROUTING a OUTPUT, které se ovšem používají k jiným než filtrovcím účelům a dále pak tabulky raw a mangle pro speciální účely modifikace a značkování paketů. Řetězec PREROUTING z tabulky nat je aktivní v době před rozhodnutím o směrování, tedy procházejí jím jak pakety určené pro firewall, tak i pro další cíle, které se nacházejí jinde v síti za firewallem. Podobně to funguje u řetězce POSTROUTING jimž protékají pakety odcházející z firewallu zároveň se směrovanými pakety. Tyto řetězce mají zvláštní význam při překladu adres NAT.

Významným prvkem rozšiřujícím iptables je stavové filtrování. Při konstrukci

filtrovacího pravidla pak je možno zohlednit stav spojení, ke kterému procházející paket patří. Rozlišujeme tyto stavy:

1. NEW - paket, který vytváří spojení a v TCP hlavičce má příznak SYN
2. ESTABLISHED - paket, který náleží některému z již vytvořených spojení.
3. RELATED - paket, který se vztahuje k nějakému probíhajícímu spojení, ale není jeho součástí.
4. INVALID - paket, který nelze přiřadit žádnému spojení a nebo má neplatný kontrolní součet.

3.2 Stavy spojení

Jednou z důležitých vlastností netfilteru je možnost sledování stavů spojení, které umožní kernelu mít přehled o všech síťových spojeních a relacích. Modul sledování spojení klasifikuje každý paket jedním ze čtyř stavů NEW, ESTABLISHED, RELATED, INVALID. První paket, který firewall vidí je klasifikován stavem NEW, odpověd na tento paket je pak klasifikována stavem ESTABLISHED a například chybové hlášení protokolu ICMP vztahující se k tomuto spojení pak bude ve stavu RELATED. Paket nepříslušející žádnému známému spojení nebo pokud se jedná o neplatný paket je klasifikován stavem INVALID.

Stav spojení se vytváří z příznaků protokolu TCP. Jestliže na firewall přichází paket TCP s příznakem SYN ACK ve druhé fázi třícestného handshake, aby potvrdil nově příchozí TCP spojení pak spojení TCP samo o sobě jestě není navázáno, ale paket v této fázi má již přiřazen stav ESTABLISHED.

Pakety přicházející na firewall se před zjištěním stavu musí defragmentovat, aby bylo možné správne rozlišit stav. Stavová tabulka pro UDP a TCP spojení je udržována v souboru

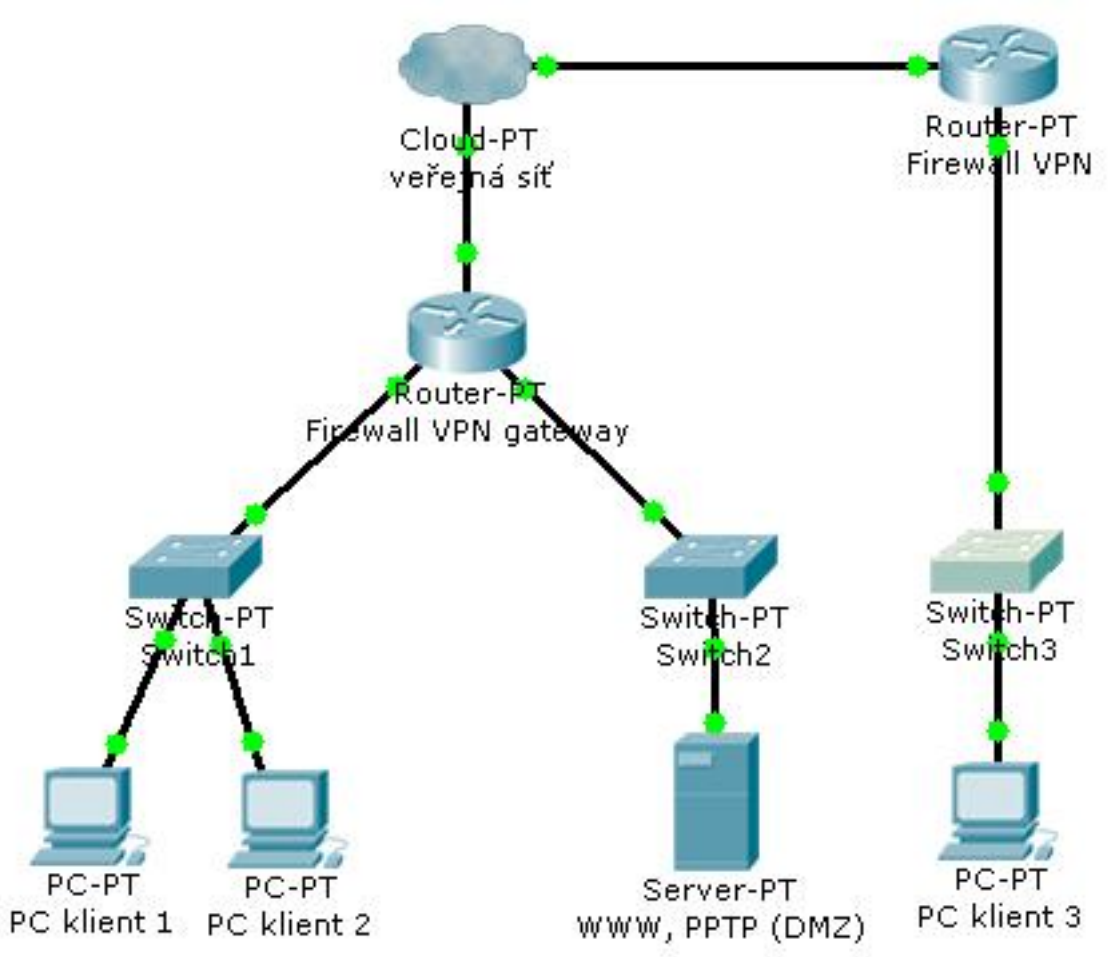
```
/proc/sys/net/ipv4/ip_conntrack
```

Při návrh firewallu se zohledňuje maximální počet spravovaných spojení a tato hodnota by měla být úměrná hardwarovým možnostem firewallu, především velikostí operační paměti.

4 NÁVRH FIREWALLU

4.1 Model sítě

Základem vytvořené modelové sítě je firewall na obr. 4.1 - Firewall VPN gateway. Tento centrální uzel spojuje přes směrovač Firewall VPN vzdálenou lokální sít s PC klient 3. Z centrálního uzlu sítě je řízena veškerá komunikace pro klientské stanice v místní síti LAN a pro server v demilitarizované zóně. Cloud-PT na obrázku představuje veřejnou síť, například internet. Jednotlivé přepínače (Switch 1-3) pouze poskytují konektivitu na linkové vrstvě a pro účely vytvoření bezpečnostní politiky firewallu nebudou dále zohledňovány.



Obr. 4.1: Schema síťě

4.1.1 Demilitarizovaná zóna

V demilitarizovaná zóně také označované zkratnou DMZ se zpravidla umísťují servery poskytující služby veřejné a nebo i vnitřní síti. DMZ musí mít na firewallu striktně nadefinována pravidla povolující příslušné směry komunikace. Z veřejné sítě jsou spojení do DMZ omezena na nezbytné minimum tak, aby byla zachována požadovaná funkcionalita. Z bezpečnostních důvodů není žádoucí umožňovat přímé spojení z DMZ do lokální sítě, tím se chrání lokální síť proti možným útokům ze strany DMZ kde by mohlo v případě špatného nastavení dojít k využití vztahů důvěry což by představovalo riziko pro lokální síť. Servery v DMZ jsou dostupné přes vyhrazené síťové rozhraní firewallu, které překládá porty služeb na cílový port a adresu serveru.

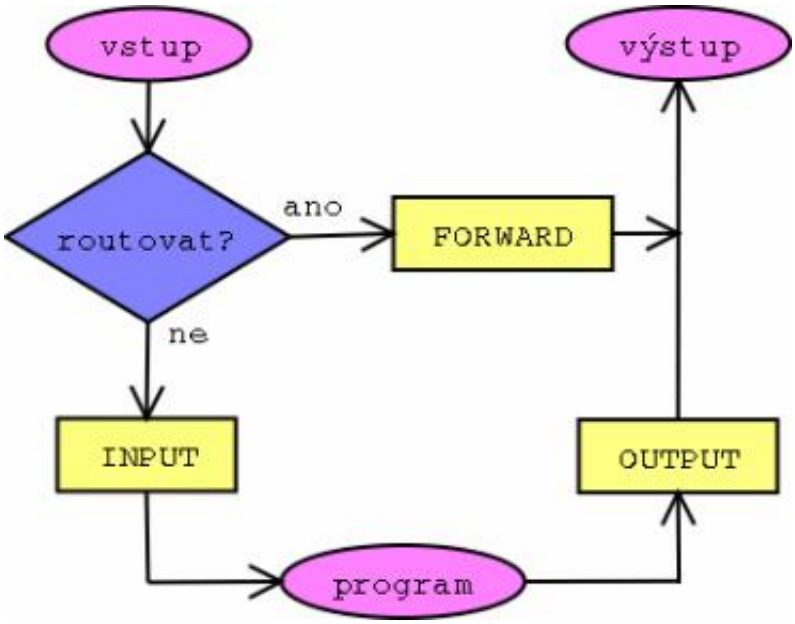
4.1.2 Lokální síť

Klientské PC jsou k centrálnímu prvku sítě připojeny přes přepínač do vyhrazeného rozhraní firewallu. Přes toto síťové rozhraní probíhá komunikace skrz firewall do cílové síťe. Firewall tedy určuje pro dané rozhraní a adresní rozsahy povolené směry komunikace. Z lokální sítě jsou dostupné jen některé služby nacházející se mimo tuto síť, zpravidla v DMZ nebo ve veřejné síti. Protože lokální síť neposkytuje žádně služby vnějšímu světu bývá komumikace ze vnějšku do místní LAN omezena pouze na spojení navázaná zevnitř, zde se využívá stavů ESTABLISHED a RELATED při definici pravidel firewallu.

4.2 Základní tabulky firewallu

Každý IP datagram s sebou nese vyjma vlastních užitečných dat také hlavičku, obsahující zejména IP adresu zdroje i adresáta, zdrojový a cílový port specifikující službu, které je datagram určen, a další informace popisující komunikaci, ke které datagram náleží. Paketový firewall je pak jakýmsi filtrem, který na základě těchto informací rozhoduje o tom, které pakety mohou být připuštěny až k cílovým programům poskytující služby, nebo které naopak smějí opustit počítač.

Každý paket, ať už pochází odkudkoliv, prochází systémem řetězců, které vytváří filtrovací tabulku uvedenou na obr. 4.2 V první fázi se snaží jádro rozhodnout, zda-li je příchozí paket určen pro tento počítač, nebo jestli je potřeba jej směrovat jinam. V případě, že cílem je firewall, předá se paket k dalšímu zpracování do řetězce INPUT. Pokud vyhoví filtrovacím pravidlům, je paket postoupen některému z lokálních procesů naslouchajícím na cílovém portu. Je-li paket určen jinému počítači



Obr. 4.2: Schema řetězců tabulky filter

a zároveň je-li firewall zkonfigurován pro směrování paketů přes síťová rozhraní, tzn. že vykonává funkci směrovače pak bude paket předán řetezci FORWARD.

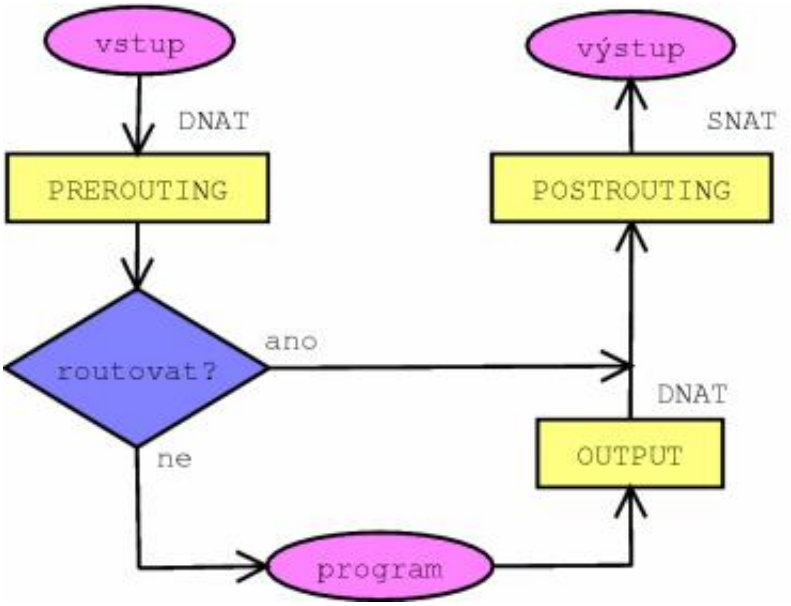
```
/proc/sys/net/ipv4/ip_forward = 1
```

Pokud je směrování paketů zakázáno paket se zahodí. Ve výchozím stavu směrování není povoleno. Pakety pocházející z některého lokálního proces odchází přes řetězec OUTPUT.

4.3 Vytváření pravidel

Pro nastavení pravidel slouží nástroj iptables (je přítomen ve verzi jádra 2.4 a vyšší). Program voláme s několika parametry. Prvním parametrem je tabulka, se kterou chceme pracovat. Pokud není specifikováno pak se jedná defaultně o tabulku filter, dále můžeme specifikovat tabulky: nat, raw, mangle. Dalším parametrem v pořadí je řetězec do kterého budeme pravidlo přidávat. Bežně jsou z tabulky filter (viz obr.4.2) dostupné tyto řetězce: INPUT, OUTPUT, FORWARD. Dále specifikujeme podmínky jež vytváří pravidlo a nakonec požadovanou akci: ACCEPT (přijmout), REJECT (zamítnout), DROP (zahodit). Akce můžeme mít uživatelsky definované, jedná se pak o skok na uživatelem specifikovaný řetězec. Další typy akcí platných v tabulce NAT (viz obr.4.3) jsou tyto: SNAT, DNAT, REDIRECT, MASQUERADE. Syntax vypadá následovně:

```
$IPT -t TABULKA ŘETĚZEC KRITÉRIA AKCE/CÍL
```



Obr. 4.3: Schema řetězců tabulky NAT

```
$IPT -t filter INPUT -p tcp --dport 80 -j ACCEPT
```

Pravidlo je určeno sadou kritérií, proti kterým se zkoumaný paket porovnává. Nevyhoví-li zkoumaný paket některé z podmínek, neuplatní se žadná akce definovaná tímto pravidlem a paket je postoupen dalšímu pravidlu. Pokud paket nevyhoví ani poslednímu pravidlu v řetězci dojde k uplatnění výchozí politiky pro daný řetězec. Jednotlivé podmínky pravidla jsou ve vztahu logického součinu, takže vyhovuje-li paket všem těmto podmínkám uplatní se na něj definovaná akce, pokuď jen jedna podmínka nevyhovuje pravidlo nemůže být pro daný paket akceptováno. V uvedeném případě je zvolena tabulka filter, paket vstupuje do řetězce INPUT. Má-li porovnávaný paket hlavičku TCP a cílový port 80 znamená to, že vyhovuje všem daným kritériím a uplatní se akce ACCEPT a přeruší se procházení řetězce pro tento paket, který je již schválen a může projít ke svému cíly.

4.4 Inicializace firewallu

Firewall je realizován sadou filtračních kritérii za příkazem iptables. Iptables se spouští pro každé pravidlo. Firewall sestává z několika desítek pravidel. Je tedy vhodné tyto pravidla vykonávat prostřednictvím nějakého bash skriptu. Spouštění jednotlivých příkazů přímo z příkazové řádky může mít za následek ne zcela očekávatelnou funkci firewallu. Je potřeba si uvědomit, že pravidla jsou aplikována v pořadí, ve kterém je definujeme v jednotlivých řetězcích a jsou přidávána buď na konec nebo na začátek řetězce dle volby za příkazem iptables. Při procházení pravidel

platí, že při první shodě paketu s pravidlem se procházení pro daný paket ukončí a z tohoto důvodu je třeba firewall psát hierarchicky, tzn. od nejkonkrétnějšího pravidla k nejobecnějšímu. Dále inicializace firewallu zpravidla vyžaduje definici konstant použitých ve skriptu, povolení kernelových modulů (v případě potřeby), vymazání existujících pravidel v řetězcích, definování výchozích politik jednotlivým řetězcům, povolení rozhraní lokální smyčky (loopback) pro správnou funkci některých lokálních služeb, nastavení adres a sítí, které chceme filtrovat.

4.5 Konstanty

Pro usnadnění práce s velkým počtem pravidel je vhodné nadefinovat si patřičné konstanty. Konstanty vytváří ve firewallu určitou symboliku. Název tak zpravidla napovídá obsah konstanty. Vhodnou volbou názvu konstanty pak dosáhneme zkrácení délky retězce což je neocenitelné při psaní pravidel a do jisté míry se tímto oprostíme od překlepů například při psaní adres nebo názvů rozhraní jelikož ty jsou za použití konstant vyjádřeny zkratkou napovídající účel. V případě změny adresy nebo názvu rozhraní se přepíše pouze daná konstanta a změna se projeví ve všech pravidlech definovaných touto konstantou. Níže uvedené definice konstant specifikují cesty k programu iptables a službě pro zavádění modulů jádra, adresy a názvy rozhraní firewallu.

```
IPT="/sbin/iptables"
MOD="/sbin/modprobe"

INET_IP="192.168.2.8"
INET_IFACE="eth0"
INET_BROADCAST="192.168.2.15"

LAN_IP="172.16.250.1"
LAN_IP_RANGE="172.16.250.0/29"
LAN_IFACE="eth1"

DMZ_IP="172.16.200.1"
DMZ_IP_RANGE="172.16.200.0/29"
DMZ_IFACE="eth2"

VPN_IP="10.0.0.1"
VPN_IP_RANGE="10.0.0.0/30"
VPN_IFACE="tun0"
```

```
LOCAL_IP_RANGE="172.16.0.0/16"

LO_IFACE="lo"
LO_IP="127.0.0.1"
```

4.6 Parametry jádra

V systému Linux se změnou nastavení procesů síťového subsystému dá ovlivnit chování netfilteru. Vhodným nastavením procesů jádra lze docílit zvýšení úrovně zabezpečení například aktivováním podpory pro kontrolu zdrojových adres rp_filter. Rychlost zpracování je vyšší než by tomu bylo v případě definování této vlastnosti za pomocí pravidel iptables. Nastavení se provádí zápisem hodnot do souborů procesů v /proc.

Omezení zpráv echo broadcast :

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

Vypnutí zdrojového směrování :

```
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
echo 0 > $f
done
```

TCP SYN cookies je mechanizmus sloužící pro rychlejší detekci a zotavení z útoků SYN flood. Tímto příkazem se aktivují SYN cookies :

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Kontrola zdrojových adres je popsána v dokumentu RFC 1812 a definuje vlastnosti pro routery IPv4. Ochranu před zfalšovanými adresami, tzn. těmi, které se na daném rozhraní nemají vyskytovat se provede tímto příkazem :

```
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
echo 1 > $f
done
```

4.7 Moduly netfilter

Vlastnosti netfilteru můžeme rozšířit zavedením dodatečných modulů. V případech kdy je požadavek na použití speciálních služeb, zvláště těch u kterých se očekává správná funkce při překladu adres, je nutné závest podpůrný modul pro tyto služby. Zpravidla protokoly pro internetovou telefonii a proprietární protokoly si žádají zavedení specializovaného modulu. Modelová síť vyžaduje zavedení modulů pro podporu protokolů GRE a PPTP.

```
$MOD ip_gre
$MOD ip_conntrack_pptp
$MOD ip_nat_pptp
```

4.8 Filtrování paketů

Při definici pravidel je třeba zajistit správné umístění nově přidávaných pravidel. Před začátkem definice pravidel se odstraní jakékoliv existující pravidla z řetězců jinak by se mohlo stát, že nová pravidla by byla přidávána za již existujících pravidla. Při porovnávání paketů vůči již existujícím pravidlům by mohla být nalezena shoda ještě před nově definovanými pravidly. Vymazaní řetězců se tedy provede níže uvedenou sadou příkazů :

```
$IPT --flush
$IPT -t nat --flush
$IPT -t mangle --flush
```

Řetězce firewallu jsou tomto okamžiku již prázdné. V případě existence i uživatelsky definovaných řetězců je nutné tyto řetězce odstranit ve všech tabulkách :

```
$IPT -X
$IPT -t nat -X
$IPT -t mangle -X
```

Firewall můžeme koncipovat dvěma způsoby. Tím prvním je povolit vše, a postupně příslušnými pravidly zakazovat nežádoucí komunikaci. Tato koncepce nepředstavuje nikterak vysoké nároky na konfiguraci a veškerá komunikace všemi směry je od základu funkční. Avšak tento přístup může snadno umožnit nežádoucí komunikaci směrem k nám nebo od nás jelikož platí co není explicitně zakázáno je povoleno.

Vzhledem k faktu, že tato varianta je méně bezpečná dále se jí nebudu zabývat. Výchozí politiku firewallu tedy nastavím na "zahoď vše co není povoleno" a to pro všechny tabulky firewallu.

```
$IPT --policy INPUT DROP
$IPT --policy OUTPUT DROP
$IPT --policy FORWARD DROP
$IPT -t nat --policy PREROUTING DROP
$IPT -t nat --policy OUTPUT DROP
$IPT -t nat --policy POSTROUTING DROP
$IPT -t mangle --policy PREROUTING DROP
$IPT -t mangle --policy OUTPUT DROP
```

Tyto příkazy nastavující výchozí politiku firewallu nejsou závislé na pozici v jednotlivých řetězcích. Výchozí politika je uplatňována pouze v případě nenalezení shody paketu s definovanými pravidly firewallu. Uplatňuje se také v případech kdy ještě neexistují pravidla v žádném z řetězců.

V první fázi se zaměřím na definici pravidel pro řetězec INPUT. Chceme-li zpřístupit službu SSH běžící na firewallu zadáme následující pravidlo.

```
$IPT -A INPUT -i $INET_IFACE --dport ssh -j ACCEPT
```

Při definici této služby je u podmínky –dport ssh použit symbolický název "ssh". Takto specifikovat službu je možné pouze v případě, že máme v souboru /etc/services definovány všechny běžně používané služby. Pak tedy je při procházení tohoto pravidla automaticky dosazeno číslo portu služby.

```
$IPT -A INPUT -i $INET_IFACE --dport ssh -m state --state NEW \
-m recent --set --name SSH -j ACCEPT
$IPT -A INPUT -i $INET_IFACE --dport ssh -m recent --update \
--seconds 60 --hitcount 4 --rttl --name SSH -j \
LOG --log-prefix "SSH_bruteforce"
$IPT -A INPUT -i $INET_IFACE --dport ssh -m recent --update \
--seconds 60 --hitcount 4 --rttl --name SSH -j DROP
```

Drobným vylepšením předchozího pravidla pro povolení SSH služby získáme efektivní obranu proti útokům hrubou silou na SSH. První pravidlo podobně jako předchozí umožňuje příchod paketů na port 22 (čili službu SSH) otevřený na firewallu. Avšak první pravidlo dále specifikuje, že je-li paket ve stavu NEW (první příchozí), je mu přiřazeno jméno SSH. Pokud se pokusíme ke službě SSH přihlásit více jak 3x za minutu, ať už úspěšně nebo ne, uplatní se pravidlo č.2 a paket je zalogován

s uvedeným prefixem. Jelikož akce LOG není terminující je ten samý paket vyhodnocem posledním pravidlem s akcí DROP. Hodnota rttl je doplňujícím kritériem. Kromě zaznamenání zdrojové adresy modulem recent se i zaznamenává hodnota TTL určující vzdálenost zdroje. Pak nemůže dojít k situaci kdy se někdo pokouší s podvrženou zdrojovou IP adresou omezit připojení k této službě. S výhodou lze tyto pravidla aplikovat i na další služby u kterých to má smysl.

Pokud nepoužíváme službu AUTH je vhodné odmítnout spojení (nikoliv zahodit) a odesílatele vyrozumět paketem s příznakem RESET . Toto pravidlo je dobré použít jelikož odesílatel je v krátké obeznámen s tím, že daná služba na firewallu neběží a tudíž nemusí čekat až mu vyprší časový limit spojení jak by tomu bylo v případě uplatnění implicitní politiky DROP.

```
$IPT -A INPUT -i $INET_IFACE -p tcp --dport auth \
 -j REJECT --reject-with tcp-reset
```

Speciálním případem jsou ICMP pakety, tedy servisní pakety používané pro přenos diagnostických a chybových zpráv. Pro správné funkci diagnostických nástrojů a také pro dodržení specifikace RFC je třeba propouštět alespoň ICMP typ 3 - "destination unreachable", "Echo request - 8" a "Time exceeded - 11", které používají diagnostické programy ping a traceroute. Přehled dalších typů ICMP lze nalézt v tab. 2.2. Pro účely specifikace ICMP paketů a pro vyšší přehlednost si nadefinuji nový řetězec ICMP_pakety. Na nově vytvořený řetězec pak odkazuji z řetězce INPUT.

```
$IPT -N ICMP_pakety
$IPT -A ICMP_pakety -p icmp --icmp-type 3 -m length -j ACCEPT
$IPT -A ICMP_pakety -p icmp --icmp-type 8 -m length \
 --length 28:92 -m limit --limit 2/s --limit-burst 5 -j ACCEPT
$IPT -A ICMP_pakety -p icmp --icmp-type 11 -m length -j ACCEPT
$IPT -A INPUT -i $INET_IFACE -p icmp -j ICMP_pakety
```

Protože chceme povolit ICMP pakety a zároveň zabránit možnému DoS útoku Ping of Death prostřednictvím ICMP protokolu použijeme u ICMP paketu typu 8 (Echo request) podmínku z modulu limit, která usměrňuje množství příchozích paketů na 2 za vteřinu a zároveň maximálně 5 po sobě jdoucích paketů, tedy celkem 10. Další vylepšení se týká maximální povolené délky ICMP paketu, kterou jsem stanovil v rozmezí 28-92 bajtů což vyhovuje běžně používaným velikostem paketů při použití diagnostického nástroje ping (Windows 32 bytů, Linux 64 bytů) a zároveň zabrání příjmu extrémně velkého paketu, který by v případě neošetřeného kernelu způsobil Ping of Death. Nutno podotknout, že stanovená horní mez velikosti ICMP

paketu zahrunuje i velikost IP záhlaví (20 bytů) a zdrojovou i cílovou adresu, která si vezme 8 bytů. Zadáme-li tedy příkaz v příkazovém řádku windows kde je explicitně uvedena velikost 65 bytů můžeme očekávat výsledek uvedený na následujícím řádku

```
D:\>ping 192.168.2.8 -l 65
Příkaz PING na 192.168.2.8 s délkou 65 bajtů:
Vypršel časový limit žádosti.
Statistika ping pro 192.168.2.8:
Pakety: Odeslané = 1, Přijaté = 0, Ztracené = 1 (ztráta 100%)
```

Firewall připojený svým rozhraním k síti LAN považuje tuto síť za důvěryhodnou, můžeme tedy povolit pakety přicházející přes rozhraní LAN. Síť DMZ podobně jako síť INET nelze považovat za zcela důveryhodnou a z tohoto důvodu se povolí přístup na firewall pouze pro spojení, která byla zahájena na firewallu. Pro správnou funkčnost meziprocesové komunikace některých aplikací a služeb je třeba povolit virtuální síťové rozhranní loopback.

```
$IPT -A INPUT -p ALL -i $LAN_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -i $VPN_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -i $LO_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -s ! $LOCAL_IP_RANGE -i $DMZ_IFACE \
-m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A INPUT -p ALL -s ! $LOCAL_IP_RANGE -i $INET_IFACE \
-m state --state ESTABLISHED,RELATED -j ACCEPT
```

Poslední dvě pravidla navíc kontrolují zdrojovou adresu, zkoumá se, zda-li paket přícházející z vnější sítě INET náhodou nemá modifikovanou zdrojovou adresu na adresu z lokální sítě LAN nebo DMZ. Praktičtější je však využít podporu přímo v jádře linuxu. Všechny ostatní nespecifikované příchozí pakety budou v souladu s implicitní politikou zahozeny.

Filtrování v řetězci OUTPUT je vhodné nastavit pro nezbytně nutné služby firewallu jež musí komunikovat s vnější sítí, aby v případě zdařeného útoku měl útočník ztížené možnosti komunikace. Řetězec OUTPUT se týká pouze komunikace vznikající na firewallu. Dle níže uvedených pravidel je tedy možná komunikace do vnějších sítí pouze službám DNS (překlad názvů na IP adresy), HTTP (pro stahování aktualizací z WWW), HTTPS, SSH (pro správa serveru a přenosy souborů SCP), WHOIS (pro identifikaci útočníka systémem PSAD).

```
$IPT -A OUTPUT -s $LO_IP -p ALL -j ACCEPT
$IPT -A OUTPUT -o $LAN_IFACE -j ACCEPT
```

```
$IPT -A OUTPUT -o $VPN_IFACE -j ACCEPT
$IPT -A OUTPUT -p icmp -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p udp --dport 53 -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --dport www -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p udp --dport https -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --dport https -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p udp --sport 1194 -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --sport ssh -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --dport whois -j ACCEPT
```

V řetězci FORWARD se musí nadefinovat povolené směrování paketů pro všechna síťová rozhraní s různým stupněm důvěryhodnosti. Nutno brát v potaz povahu komunikace a nastavit příslušná pravidla pro komunikaci oběma směry. Směrování paketů vstupujících na rozhraní INET_IFACE a dále putující na rozhraní LAN_IFACE nebo DMZ_IFACE bude umožněno jen těm, jejichž spojení bylo navázáno zevnitř, tzn. z LAN nebo DMZ. Jiný případ je kdy chceme povolit služby uvnitř DMZ. Pak se tedy definuje stav NEW a specifikují se povolené porty v daném směru.

```
$IPT -A FORWARD -i $LAN_IFACE -j ACCEPT
$IPT -A FORWARD -i $VPN_IFACE -j ACCEPT
$IPT -A FORWARD -i $DMZ_IFACE -o $INET_IFACE -j ACCEPT
$IPT -A FORWARD -i $DMZ_IFACE -o $LAN_IFACE -m state \
 --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $INET_IFACE -o $LAN_IFACE -m state \
--state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $INET_IFACE -o $DMZ_IFACE \
-m state --state ESTABLISHED,RELATED -j ACCEPT

$IPT -N dmz_forward
$IPT -A FORWARD -i $INET_IFACE -o $DMZ_IFACE -m state \
--state NEW -j dmz_forward
$IPT -A dmz_forward -p tcp --dport www -j ACCEPT
$IPT -A dmz_forward -p tcp --dport https -j ACCEPT
$IPT -A dmz_forward -p udp --dport https -j ACCEPT
$IPT -A dmz_forward -p gre -j ACCEPT
$IPT -A dmz_forward -p tcp --dport 1723 -j ACCEPT
```

Pro větší přehlednost a snažší pochopení uvádím povolené směry komunikace v tab. 4.1 Drobnou zvláštností je zadaní protokolu GRE. Ve zkratce, jedná se o zvláštní tunelovací protokol s číslem 47 navržený pro zapouzdření paketů síťové vrstvy.

Tab. 4.1: Přehled směrování.

Zdrojová síť	**Cílová síť**	**Popis**
LAN	*	Ze sítě LAN kamkoliv
VPN	*	Ze sítě VPN kamkoliv
DMZ	INET	Bez omezení
DMZ	LAN	Pouze pro navázaná spojení z LAN
INET	LAN	Pouze pro navázaná spojení z LAN
INET	DMZ	Pro navázaná spojení z DMZ a nová spojení na port 80, 443 a 1723

4.9 Překlad adres

Nyní se dostáváme z oblasti, filtrační do oblasti překladu adres. Kromě filtrovací tabulky mají iptables ještě další tabulku, kterou procházejí pakety a sice NAT (Překlad síťových adres). Stejně jako filtrovací tabulka, i tabulka NAT obsahuje tři řetězce, které se však nepoužívají k filtrování nýbrž k modifikaci adres paketů. Princip je následující. Pokud paket při průchodu vyhoví zadanému pravidlu, tak mu je podle určeného vzoru změněna adresa jeho odesílatele resp. příjemce podle určeného vzoru. Podle toho hovoříme buď o překladu adres odesílatele SNAT (Zdrojový překlad síťových adres)) nebo překladu adres příjemce DNAT (Překlad cílových síťových adres). Přichází-li paket, nejprve projde řetězcem PREROUTING, kde mu můžeme změnit adresu příjemce, tedy DNAT. Odchází-li paket, prochází řetězcem POSTROUTING kde mu může být změněna zdrojová adresa. Znázorněno na obr.4.3

4.9.1 SNAT

SNAT provádí maskování IP adresy směrovaných paketů na adresu vnějšího rozhraní firewallu. Překlad zdrojových adres se používá máme-li lokální síť s neveřejnými ip adresami a potřebujeme přistupovat na internet přes veřejnou ip adresu nebo chceme-li z nějakého důvodu zamaskovat zdrojovou ip adresu. Proto veřejnou adresu přidělíme routeru a ostatním počítačům přidělíme privátní IP adresy, které jsou k tomuto účelu rezervovány podle RFC a do NAT tabulky routeru vložíme následující pravidlo.

```
$IPT -t nat -A POSTROUTING -o $INET_IFACE -j SNAT --to-source $INET_IP
```

Pravidlo způsobí, že pokud z vnitřní sítě přijde paket, který má opustit router přes vnější rozhraní INET , tak dojde k nahrazení jeho původní IP adresy za adresu

INET_IP. Příjemce paketu pak uvidí vnější adresu routeru namísto lokální adresy. Aby se pakety v opačném směru dostaly zpět do lokální síťě provede se automaticky v tabulce mapování na původní adresu a port.

Podobně jako v předchozím případě lze zdrojovou adresu zamaskovat použitím tzv. maškarády, kdy je zdrojová adresa změněna automaticky na adresu výstupního rozhraní, v tomto případě na adresu rozhraní INET_IP.

```
$IPT -t nat -A POSTROUTING -o $INET_IFACE -j MASQUERADE
```

Obě definice fungují téměř stejně ovšem s tím rozdílem, že v případě maškarády se posuzuje toto pravidlo pro každý jednotlivý odchozí paket což se projeví navýšením režije při překladu adres, ale na druhou stranu pokud je adresa výstupního rozhraní nastavována z DHCP serveru pak se provede překlad na výstupní adresu automaticky. V případě více veřejných (nebo i privátnich) adres lze provést překlad na rozsah těchto adres a zároveň dochází k vyvažování záteže jelikož se odchozí spojení dynamicky překládá na jednu z těchto adres.

```
$IPT -t nat -A POSTROUTING -o $INET_IFACE \
-j SNAT --to-source 1.2.3.4-1.2.3.6
```

4.9.2 DNAT

DNAT použijeme tehdy, je-li potřeba změnit IP adresu adresáta paketu. V demilitarizované zóně se nachází webový server a VPN server, který má být dostupný z vnější sítě. Provede se překlad na cílovou adresu a příslušné porty.

```
$IPT -t nat -N dmz_prerouting
$IPT -t nat -A PREROUTING -i $INET_IFACE -j dmz\_prerouting
$IPT -t nat -A dmz_prerouting -p tcp --dport www -j DNAT \
--to 172.16.200.2
$IPT -t nat -A dmz_prerouting -p tcp --dport 1723 -j DNAT \
--to 172.16.200.2
$IPT -t nat -A dmz_prerouting -p gre -j DNAT --to 172.16.200.2
```

Touto definicí se zajistí přesměrování každého paketu cíleného na rozhraní INET_IP a zároveň s cílovým portem 80 na adresu WWW serveru umístěného v DMZ. Podobně je tomu i u mapování portu 1723 pro službu VPN, která navíc vyžaduje i dodatečné mapování protokolu GRE. Je potřeba brát na vědomí skutečnost, má-li paket být cílen na adresu v DMZ je potřeba mít odpovídající pravidlo pro povolení komunikace na DMZ v řetězci FORWARD tabulky filter.

Tab. 4.2: Výchozí hodnoty TTL operačních systémů.

Hodnota	**Operační systém**
64/65	FreeBSD
60	Irix
64	Linux
60	MacOS
255	Solaris
32	Windows 95/NT
128	Windows 98/2000/XP

4.10 Modifikace paketů

Další zajímavou a neméně důležitou je i tabulka mangle, která nabízí tyto možnosti. MARK slouží pro nastavení značkovacích hodnot, které jsou přidružené se specifickými pakety. MARK hodnoty mohou být používány spolu s pokročilými routovacími schopnostmi v Linuxu, např. pro omezování toku dat (shaping), měření toku dat (accounting), apod.

TOS – typ služby, reprezentováno osmibitovou hodnotou, která udává jak mají být pakety doručeny. Zda mají být doručeny přednostně (s minimální odezvou), s maximální propustností či pro označování neprioritního provozu. S volbou –set-tos můžeme předat tyto parametry: Minimize-Delay, Maximize-Throughput, Maximize-Reliability, Minimize-Cost a Normal-Service. Nutno podotknout, že s nastavením TOS se nemusí každý router na internetu vyrovnat, v lepším případě může dojít jen k ignorování hodnoty TOS. Implementace TOS je spíše vhodná pro větší LAN nebo WAN sítě, kde se s využitím prioritizace datových toků počítá.

Dalším rošířením je cíl TTL, hodnota doby životnosti paketu. Nabízí možnost pro změnu hodnoty TTL v IP hlavičce paketu na stejnou hodnotu na všech odcházejících paketech. Důvod, proč této možnosti využít je více. Nastavením hodnoty TTL pro odchozí pakety z firewalu lze zabránit ISP ve zjištění, že v síťi za firewallem se nachází vícero počítačů. Dalším důvodem proč nastavit hodnotu TTL může být ztížení identifikace použitého operačního systému na firewallu.

Pro identifikaci použitého operačního systému stačí když od cíle našeho zájmu přijde jakýkoliv paket, který lze zachytit například síťovým analyzátorem Wireshark. Z hlavičky paketu se zjistí hodnota TTL, která bude například 50. Poté se určí kolik routerů stojí na cestě mezi námi a cílem, přepokládejme hodnotu 14. K tomuto účelu dobře poslouží utilitka traceroute. Obě čísla se sečtou a získáme tak výchozí hodnotu TTL, kterou použil operační systém při odeslání paketu. V tomto případě 64. Na konec už stačí tuto hodnotu pouze porovnat s uvedenou tabulkou 4.2 a vybrat ten

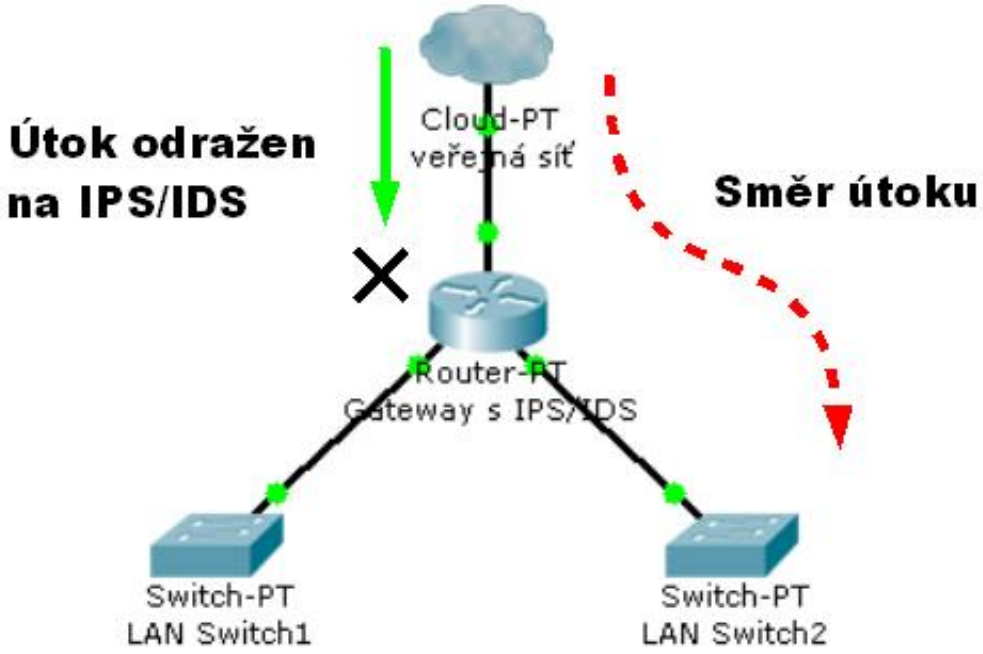
správný operační systém nebo můžeme alespoň zúžit okruh těch možných.

S využitím poznatků předchozího odstavce můžeme napsat pravidlo, které v tabulce mangle mění hodnotu TTL pro všechny pakety odcházející z vnějšího síťového rozhraní.

```
$IPT -t mangle -A POSTROUTING -o $INET_IFACE -j TTL --ttl-set 70
```

5 SYSTÉMY IDS/IPS

Sítě a počítačové systémy byly dříve běžně chráněny hraničními obranými prvky routery s ACL a firewaly. Pokusy o útok na síť nebo cílový systém byly detekovány buď systémy NIDS nebo systémy HIDS. Tyto typy zařízení poskytují silnou a robustní obranu, avšak až technologický pokrok umožnil vzájemnou integraci těchto systémů. IPS systém kombinuje vlastnosti firewallu a IDS, který nejenom detekuje útoky, ale dokáže jim i zabránit jak je znázorněno na obr. 5.1.



Obr. 5.1: Směr útoku a místo terminace

Pod pojmem aktivní reakce se skrývají metody umožňující dynamickou rekonfiguraci nebo úpravu síťových přístupových mechanismů, relací nebo dokonce jednotlivých paketů na základě poplachů generovanzých z IDS. Aktivní reakce systému IDS nastává až po výskytu události, takže například jeden paket nebo i více paketů sledovaného útočného spojení úspěšně projde na první pokus a při dalších pokusech dojde k zablokování. Zatímco zařízení implementující aktivní reakci jsou užitečné, výše zmíněný aspekt je dělá nevyhovujícím řešením pro komplexní zabezpečení sítě. Systémy prevence průniku bývají v síti začleněny jako tzv. in-line jednotky , které prověřují pakety a činí rozhodnutí před jejich směrováním k cíli. Tento typ zařízení má schopnost zabránit jednotlivým škodlivým paketům již při prvním pokusu blokováním nebo modifikováním tohoto útočného paketu.

Přehled technologií prevence průniku:

1. **Ochrana systémové paměti a procesů**
   Tento typ strategie prevence průniku sídlí na systémové úrovni. Ochrana paměti sestává z mechanismu jež zabraňuje procesu narušit paměť jiného procesu na témže systému. Procesová ochrana obsahuje mechanismus pro sledování spouštění procesů se schopností ukončit procesy, které jsou podezřelé z útočných aktivit.

2. **Odstřihnutí relace**
   Tato strategie prevence narušení ukončuje TCP relace zasláním TCP RST paketu oběma stranám spojení. Je-li detekován pokus o útok, dojde k odeslání TCP RST paketu. Aby byla tato metoda efektivní je důležité použít pro ukončení relace správná pořadová a potvrzovací čísla.

3. **Zařízení spolupracující s bránou**
   Tento typ dovolí detekčnímu zařízení dynamicky spolupracovat se síťovou bránou - routerem nebo firewallem. Když je detekován pokus o útok, detekční zařízení může nařídit routeru nebo firewallu zablokování tohoto spojení

4. **Inline síťové zařízení**
   In-line je takové zařízení, které je přímo v cestě mezi komunikujícími stranami a zároveň na vhodném místě komunikace jež má schopnost pozměnit a blokovat útočné pakety, které prochází přes síťová rozhraní. Výsledek je pak stejný jako router nebo firewall kombinovaný s IDS se schopnostmi analýzy na základě vzorků. Detekce a reakce probíhá v reálném čase a předtím než je paket odeslán na cílovou síť.

5.1 Slabá místa IDS/IPS

K technologiím prevence průniku a aktivní reakce se váže několik rizik. Při některých událostech může být zcela legitimní datový provoz zobrazován jako útok s podobnými charakteristikami škodlivého datového provozu. Tato situace může vzniknout při porovnávání s nevhodnými vzorky. Chybné vyhodnocení může vést ke vzniku DoS při zcela legitimním datovém provozu. Navíc, pokud útočník zjistí toto chování systému nebo má podezření, že jsou použity metody prevence průniku pak může účelně vytvořit DoS útok proti sítí s podvrženými zdrojovými IP adresami. Riziko DoS útoku lze snížit použijeme-li vylučovací seznamy tzv. whitelisty. Whitelist obsahuje seznam síťových zdrojů, které nebudou nikdy blokovány. Do whitelistů je dobré

zahrnout systémy DNS, mail servery, routery, firewally a ostatní zařízení kritické pro chod sítě.

Další problém je spojen s nasazením systémů se schopností ukončovat relace. Když tyto systémy ukončují relace RST pakety, útočník pak z těchto paketů může získat nejenom informaci o nasazení IPS, ale také může zjistit o jaký typ systému se jedná. Existují snadno dostupné nástroje vykonávající pasivní identifikaci operačního systému na základě analýzy paketů. Tento typ informace umožní útočníkovi vyhnout se IPS nebo provést přímo útok na IPS. Problém může také nastat s IDS systémy jež vyžadují spolupráci s bránou. Mějme tuto situaci, kdy detekční zařízení nařídí routeru nebo firewallu blokovat pokus o útok, avšak díky síťové latenci útok již prošel bránou ještě před přijetím povelu z detekční jednotky.

5.2 Obsahové filtrování a inspekce

Síťové systémy IDS/IPS nasazované na IP sitích musí být schopné kontrolovat obsah paketů a všechna pole z hlavičky paketu za účelem detekce charakteristik, které mohou představovat nekalou činnost. Netfilter v linuxovém jádře generuje dostatečně detailní logy a spolu s rozšířením pro porovnávání řetězců může dokonce prohledávat a generovat logy pro vzorky v aplikační části IP paketů.

V této části mé práce představím dva programy - FWSNORT a PSAD a budu se zabývat jejich efektivní implementací na modelovou síť. Oba programy dohromady jsou schopny pracovat přibližně s 8000 pravidly pocházejících z IDS systému SNORT.Cílem FWSNORTu a PSADu je vhodně rozšířit funkcionalitu iptables o systém IPS a IDS. Avšak tyto zmíněné programy by měly být spíše doplňkem již existujícího systému IDS než komplexním řešením pro velké sítě. Pro nasazení v menší síti jakou je daná modelová síť jsou však tyto programy plně dostačující a splní svůj účel.

Iptables mají schopnost logovat téměř jakékoliv z polí síťových a transportních hlaviček. Nutno podotknout, že ne všechny pole jsou logovány ve výchozím stavu, ale lze toho docílit vhodným parametrem k příkazu iptables. Pro ilustraci uvádím schopnosti logování iptables ve výchozím stavu, níže jsou zobrazeny tři příklady log zpráv, které byly generovány TCP SYN paketem, UDP paketem a ICMP paketem echo request. Všechny tři pakety byly zalogovány v řetězci FORWARD.

TCP syn:

```
May 08 10:27:11 phobos kernel: retezec FORWARD: IN=eth0 OUT=eth2
SRC=192.168.2.10 DST=172.16.250.2 LEN=60 TOS=0x10 PREC=0x00 TTL=63
ID=56341 DF PROTO=TCP SPT=37481 DPT=6776 WINDOW=5840 RES=0x00 SYN URGP=0
```

UDP:

```
May 08 10:31:13 phobos kernel: retezec FORWARD: IN=eth0 OUT=eth2
SRC=192.168.2.10 DST=172.16.250.2 LEN=28 TOS=0x00 PREC=0x00 TTL=63
ID=26893 PROTO=UDP SPT=46564 DPT=20433 LEN=8
```

ICMP:

```
May 08 10:33:17 phobos kernel: retezec FORWARD: IN=eth0 OUT=eth2
SRC=192.168.2.10 DST=172.16.250.2 LEN=84 TOS=0x00 PREC=0x00 TTL=63
ID=0 DF PROTO=ICMP TYPE=8 CODE=0 ID=34653 SEQ=0
```

Význam jednotlivých polí v ukázkovém příkladu je následující:

- retezec FORWARD - pravidlo, které generovalo tuto log zprávu mělo nastavenou volbu –log-prefix
- IN a OUT rozhraní firewallu přes jimiž dané pakety prošly
- zdrojová a cílová adresa
- další pole mají následující význam: LEN= délka, TOS= typ služby, TTL= hodnota time-to-live, ID= zde je uveden offset fragmentu,DF= příznak fragmentace, PROTO= protokol
- TCP a UDP pakety také zahrnují zdrojové a cílové porty

Pole specifická pro jednotlivé protokoly:

- TCP: WINDOW= velikost okna, RES= rezervní bity, SYN= příznak TCP, URGP= ukazatel naléhavosti
- UDP: LEN= délka záhlaví UDP
- ICMP: TYPE= typ ICMP paket, CODE= kód ICMP paketu, ID, SEQ= pořadové číslo

Výše uvedený text specifikuje strukturu log zpráv, se kterými pracuje IDS systém PSAD. Náhled do aplikační vrstvy je poněkud složitější, avšak je základem analýzy datového provozu pravidly SNORT. Netfilter nabízí rozšíření v podobě modulu pro analýzu řetězců v aplikační části IP paketů. Stejně jako u všech ostatních rozšíření netfilteru najdeme i zde dvě hlavní komponenty - kernelový modul (může být zakompilován přímo v jádře) a rozhraní pro uživatelské prostředí jež je zakompilováno do knihoven, které používá netfilter. Kernelová část rozšíření pro analýzu řetězců je zodpovědná za vykonávání Boyer-Moore vyhledávání v datové části IP paketu. Uživatelské rozhraní přebírá řetězec z příkazové řádky a přidá ho do datových struktur jádra.

5.3 FWSNORT

FWSNORT překládá SNORT pravidla do ekvivalentních iptables pravidel a sestavuje bash skript pro implementaci výsledné politiky do iptables. FWSNORT rozšiřuje netfilter o další dvě funkce. Jednak záplatuje uživatelské rozhraní iptables a tím přidává možnost analýzy paketů dle kritéria –hex-string a dále nabízí možnost analyzovat existující sadu pravidel iptables. Použijeme-li volbu –snort-sid pak lze FWSNORTem překládat jednotlivá SNORT pravidla do ekvivalentních iptables pravidel. Tato volba je užitečná pro určení zda bude iptables akceptovat datový provoz shodující se s nějakým SNORT pravidlem.

Ačkoliv iptables mohou manipulovat s několika tisíci pravidly, bylo by neefektivní zahrnout pravidla pro datový provoz, který firewall nepovoluje. Například dejme tomu, že máme webový server v interní síti chráněn firewallem s iptables, dále předpokládejme, že iptables byly konfigurovány tak, aby nepovolily jakýkoliv externí přístup k webovému serveru přes port 80. V této situaci by bylo bezúčelné překládat SNORT pravidla pro analýzu webových útoků do detekčních pravidel iptables. Přece jenom by iptables měly být konfigurovány jen pro datový provoz jdoucí do firewallu nebo skrz něj, který je nezbytný pro funkci sítě a veškerý ostatní provoz by měl být logován a zahozen.

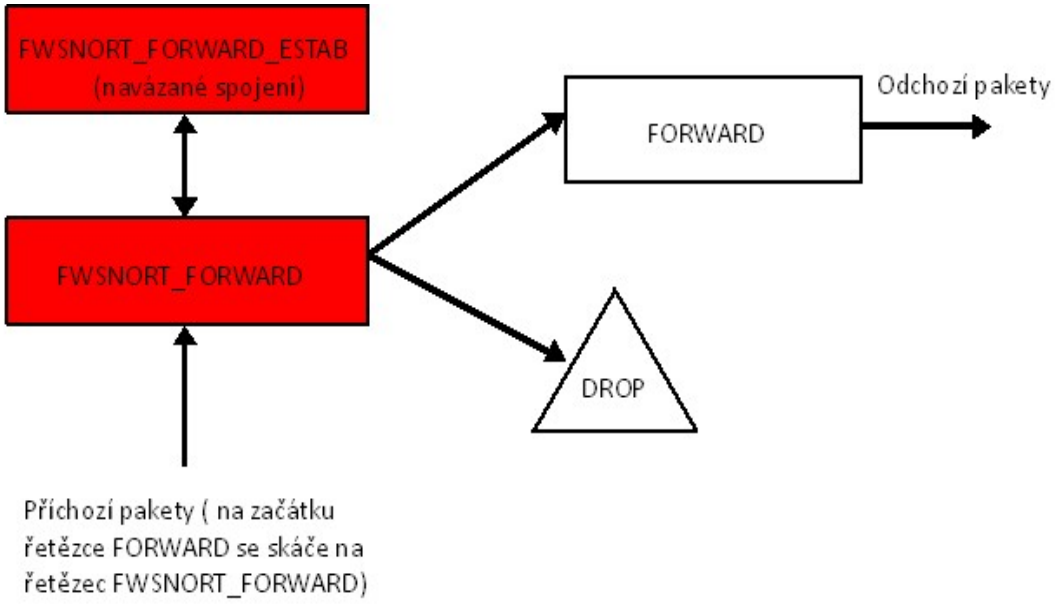
5.3.1 FWSNORT.SH

FWSNORT generuje bash skript, který obsahuje všechny příkazy iptables nezbytné pro implementaci politik. Ve výchozím nastavení iptables jen logují datový provoz odpovídající pravidlům SNORT, avšak FWSNORT může také sestavit politiku, která nařizuje iptables zamítnout takový datový provoz ihned po zápisu logu. Parametrem –ipt-reject za příkazem fwsnort se docílí sestavení pravidel se schopností ukončit spojení v závislosti na použitém protokolu. Pro nestavové protokoly UDP a ICMP se při ukončení spojení generuje ICMP zpráva s kódem 3 port unreachable viz tab. 2.1.

Pro UDP a ICMP pakety tato forma ukončení spojení nepředstavuje žádný problém, protože tyto protokoly jsou nestavové a takto zamítnuté spojení nezpůsobí opětovný přenos paketů. Avšak, toto jednoduché řešení nelze aplikovat pro TCP kde každý paket v sestavené relaci je potvrzován a chybějící pakety v toku jsou opětovně vyslány. Pro TCP pakety používá FWSNORT parametr pro iptables –reject-with tcp-reset, takže například paket obsahující řetězec, který má způsobit přetečení vyrovnávací paměti na interním webovém serveru je poslán přes rozhraní firewallu a následně je celá relace shozena firewallem před tím než útok vůbec dosáhne web serveru.

V závisloti na počtu rozhraní firewalu FWSNORT vytváří a přidává iptables pravidla až pro pět uživatelsky definovaných řetězců. FWSNORT může manipulovat až se třemi rozhraními: jedním externím, jedním interním a DMZ.

Tok paketů pro řetězec FORWARD je uveden na obr. 5.2



Obr. 5.2: Začlenění řetězců FWSNORT do iptables

5.3.2 Implementace FWSNORT

Instalace FWSNORT je triviální a proto se nebudu zabývat instalací samotnou, nýbrž zmíním pouze požadavky na systém pro správnou funkci aplikace. FWSNORT vyžaduje rozšíření iptables o string-matching podporu. Tato podpora je přítomna od verze iptables 1.2.9 a verze linuxového jádra 2.6.14 nebo vyšší. FWSNORT vyžaduje práva superuživatele root a spouští se příkazem fwsnort. Po spuštění příkazu se provede analýza pravidel SNORT a zjistí se aplikovatelnost pravidel na daný systém a pravidla uvedená v tabulkách a řetězcích iptables. Volbou –strict za příkazem fwsnort docílíme překladu pouze těch pravidel, které lze přeložit na již definovaná pravidla iptables. V konfiguračním souboru /etc/fwsnort/fwsnort.conf je třeba definovat konstantu HOME_NET a nastavit ji adresu lokální sítě 172.16.0.0/16. Tímto je systém FWSNORT připraven k integraci s firewallem. Níže uvedený příkaz fwsnort s parametry se integruje do skriptu konfigurace firewallu.

```
fwsnort --strict --ipt-reject --ipt-apply
```

Parametr –ipt-reject zajistí přidání pravidel pro ukončení spojení, které obsahují vzorky škodlivého kódu v aplikační části datového paketu. Parametr –ipt-apply

provede přidání pravidel z vygenerovaného skriptu v /etc/fwsnort/fwsnort.sh do iptables. Po spuštění příkazu s uvedenými parametry je již systém IPS plně funkční a připraven k inspekci datového provozu.

```
debian:~# fwsnort --strict
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
    Snort Rules File          Success   Fail      Ipt_apply Total

[+] attack-responses.rules    10        7         0         17
[+] backdoor.rules            60        16        30        76
[+] bad-traffic.rules         9         3         2         12
[+] bleeding-all.rules        291       5343      165       5634
[+] chat.rules                5         25        4         30
[+] ddos.rules                18        14        5         32
[+] dns.rules                 15        6         15        21
[+] dos.rules                 6         10        5         16
[+] experimental.rules        0         0         0         0
[+] exploit.rules             33        49        26        82
[+] web-attacks.rules         0         46        0         46
[+] web-cgi.rules             0         350       0         350
[+] web-client.rules          0         25        0         25
[+] web-coldfusion.rules      0         35        0         35
[+] web-frontpage.rules       0         35        0         35
[+] web-iis.rules             5         114       5         119
[+] web-misc.rules            26        302       26        328
[+] web-php.rules             2         124       2         126
[+] x11.rules                 2         0         2         2
                              =======================================
                              755       7715      471       8470

[+] Generated iptables rules for 755 out of 8470 signatures: 8.91%
[+] Found 471 applicable snort rules to your current iptables
    policy.
```

Pravidla generované aplikací FWSNORT a uložené do souboru fwsnort.sh obsahují signatury popisující charakteristiky útoku. Paket jdoucí do lokální sítě se zkoumá v aplikační části proti všem pravidlům z fwsnort.sh. Inspekce je provedena algoritmem Boyer-Moor a v tomto případě se jedná o signaturu FTPON, viz druhý řádek ukázky.

```
$IPTABLES -A FWSNORT_OUTPUT_ESTAB -d 172.16.0.0/16 -p tcp --dport 666
-m string --string "FTPON" --algo bm -m comment --comment "sid:157;
msg:BACKDOOR BackConst ruction 2.1 Client FTP Open Request;
classtype:misc-activity; rev:5; FWS:1.0.4;" -j LOG --log-ip-options
--log-tcp-options --log-prefix "[5] SID157 ESTAB "
```

5.4 PSAD

PSAD sestává ze sady třech systémových démonů (psad, kmsgsd, psadwatchd), kteří analyzují log zprávy z iptables, aby mohli zjistit případné skenování portů, zkoušení exploitů, apod. Mezi přednosti PSAD patří e-mailové varovné hlášení, DShield poplachy, automatické blokování IP adres ze kterých probíhá pokus o útok. Automatické blokování IP není obecně příliš dobrý nápad, protože chytrý útočník pravděpodobně rychle zjistí přítomnost této funkce a může pak provádět útoky nebo skeny s podvrženými IP adresami sítí se kterými komunikuje lokální nebo DMZ síť a tím způsobí útok DoS jelikož se tyto podvržené IP adresy přidají do seznamu blokovaných IP a znemožní komunikaci s těmito cíly legitimním uživatelům.

Jakmile iptables vygenerují log zprávy, psad použije skórovací mechanismus, aby přiřadil úroveň nebezpečí těmto skenům. Tato úroveň je odvozena z počtu paketů logovaných iptables, rozsahem skenovaných portů a tomu zda-li skeny odpovídají nějakým vzorkům TCP, UDP nebo ICMP, které jsou automaticky přiřazené určité úrovni nebezpečí. Provádíme-li kompletní logování TCP příznaků v iptables zprávách, pak může PSAD detekovat i tyto skeny: TCP connect(), SYN(napůl otevřené), FIN, NULL, XMAS, SYN/FIN, nmap a OS fingerprinting. Protože skenování portů zpravidla neobsahuje žádná aplikační data může PSAD detekovat všechny tyto skeny bez toho aniž by musel spolupracovat s aplikací FWSNORT.

5.4.1 Architektura PSAD

Během instalace PSAD přenastaví syslogd démona tak, aby zapisoval všechny zprávy kern.info do pojmenované roury umístěné ve /var/lib/psad/psadfifo. Kmsgsd démon je odpovědný za otevření této roury, čtení log zpráv z iptables a zapisování všech takových zpráv do datového souboru psad v /var/log/psad/fwdata kde jsou tyto záznamy analyzovány PSADem. Tímto způsobem je PSADu dodáván tok dat, který obsahuje pouze zprávy jež vygenerovaly iptables. Démon psadwatchd se stará o nepřetržitý chod démonů psad a kmsgsd.

PSAD poskytuje různé volby jako parametry příkazové řádky psané za PSAD. Asi nejužitečnějším příkazem je psad –Status, jež zobrazí souhrn všech IP adres, které přímo skenovaly síť nebo firewall. Výstup obsahuje informace o využívání

systémových prostředků jednotlivými démony, celkový počet zpracovaných iptables zpráv a tabulku, která vypisuje všechny zdrojové IP adresy, ze kterých bylo skenováno zároveň s počtem paketů TCP, UDP a ICMP, dále iptables řetězec a fyzické rozhraní na kterém byl sken detekován. Více v kapitole 7.

5.4.2 Implementace PSAD

Systém PSAD se instaluje obdobným způsobem jako FWSNORT, ale narozdíl od FWSNORTu vyžaduje po instalaci drobnou úpravu systémového nastavení pokud instalace neproběhne správně. Velice důležité je uvědomit si, že PSAD využívá pro analýzu paketů log zprávy generované iptables. Kdyby v pravidlech firewallu neexistovalo jediné pravidlo, které provádí akci LOG, pak by to znamenalo nefunkčnost PSAD. Obvykle je nejjednodušší a nejbezpečnější způsob konfigurace firewallu v první řadě povolit nezbytné minimum datového provozu a poté definovat pravidlo "zahoď a loguj" na konci sady pravidel. Instalátor PSAD provede prvotní nastavení systému automaticky. Po instalaci je nutné ověřit přítomnost řetězce

```
kern.info |/var/lib/psad/psadfifo
```

v souboru /etc/syslog.conf. PSAD narozdíl od FWSNORTu nabízí nepřeberné množství konfiguračních možností. Zde uvádím ty nejdůležitější parametry s krátkým komentářem.

```
#Definice lokalni site
HOME_NET                    172.16.0.0/16;
#Definice externich siti
EXTERNAL_NET                any;
#Prohledavani log zprav iptables se vsemi prefixy
FW_SEARCH_ALL               Y;
#Demon, ktery se stara o logovani
SYSLOG_DAEMON               syslogd;
#Urovne citlivosti IDS, vyjadreno poctem paketu
DANGER_LEVEL1               5;
DANGER_LEVEL2               15;
DANGER_LEVEL3               80;
DANGER_LEVEL4               500;
DANGER_LEVEL5               4000;
#Interval kontroly souboru s novymi daty z syslog demona
CHECK_INTERVAL              5;
#Pocet zkousenych portu z externi site, hodnota 10 udava
```

```
#po kolika zkousenych portech je akce vyhodnocena jako
#skenovani portu, v tomto pripade 11 (0-10)
PORT_RANGE_SCAN_THRESHOLD   10;
#Zapis pouze do systemoveho logu
ALERTING_METHODS            noemail;
#Maximalni hodnota TTL
MAX_HOPS                    25;
#Minimalne uroven nebezpeci od ktere se zahajuje sledovani zdroje
#utoku a zaznamenavani dat, prip. take odeslani varovne zpravy
MIN_DANGER_LEVEL            2;
#Zapina schopnost IDS nastavit blokani zdroje utoku/skenu
ENABLE_AUTO_IDS             Y;
#Uroven pri, ktere se aplikuji pravidla blokovani
AUTO_IDS_DANGER_LEVEL       5;
#Doba po, ktere se blokovani zdroje utoku/skenu odstrani
AUTO_BLOCK_TIMEOUT          3600;
#Metoda blokovani zdroje, v tomto pripade iptables, ale
#lze i pouzit tcpwrappers
IPTABLES_BLOCK_METHOD       Y;
#Retezce pro iptables, v nich se provadi blokovani na pokyn IDS
IPT_AUTO_CHAIN1             DROP, src, filter, INPUT,
 1, PSAD_BLOCK_INPUT, 1;
IPT_AUTO_CHAIN3             DROP, both, filter, FORWARD,
 1, PSAD_BLOCK_FORWARD, 1;
#Vymazat pravidla nastavena PSADem po po startu PSAD
FLUSH_IPT_AT_INIT           Y;
#Kontrola pritomnosti retezcu PSAD v iptables
IPTABLES_PREREQ_CHECK       1;
#Zahodit soubor fwdata s logovacimi zaznamy po spusteni PSAD
TRUNCATE_FWDATA             Y;
#Kontrola behu vsech demonu PSAD, kazdych 5 vterin a maximalne
#10x pred tim nez se vykona restart demonu
PSADWATCHD_CHECK_INTERVAL   5;
PSADWATCHD_MAX_RETRIES      10;
```

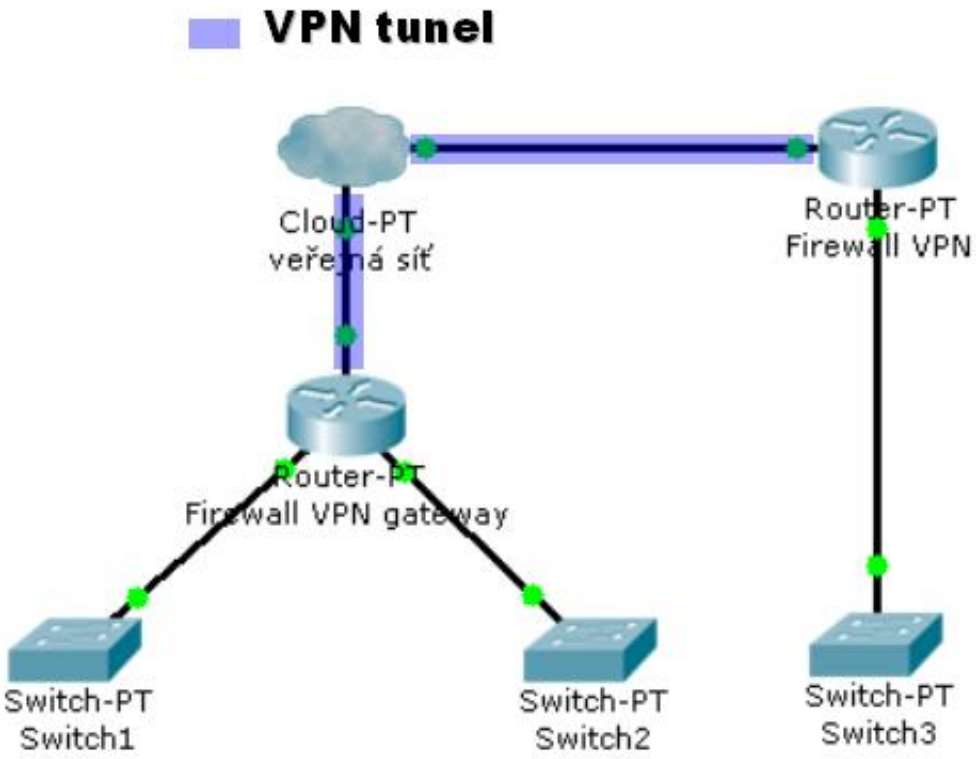
Parametry pro konfiguraci byly zjištěny empiricky a na základě posouzení charakteru komunikace v modelové síti. Kompletní nekomentovaný konfigurační soubor je v příloze C.

Aby vše fungovalo přinejmenším tak, jak bylo popsáno výše je třeba umístit tyto dvě pravidla do skriptu pro konfiguraci firewallu. Kompletní skript firewallu se nachází v příloze A.

```
$IPT -A INPUT -j LOG --log-prefix "retezec INPUT: " --log-tcp-options \
--log-tcp-sequence --log-ip-options
$IPT -A FORWARD -j LOG --log-prefix "retezec FORWARD: " --log-tcp-options \
--log-tcp-sequence --log-ip-options
```

6 VPN

VPN je zkratkou pro virtuální privátní síť - virtual private network. Jedná se o počítačovou síť, která využívá fyzické komponenty existujících privátních sítí a veřejných sítí a existuje tedy pouze virtuálně. Nespornou výhodou VPN je šifrování datového přenosu což znemožní nebo velmi zkomplikuje odposlech dat během přenosu po veřejné síti. Trasa virtuálního tunelu aplikovaného na modelovou síť je vyznačena na obr. 6.1 Jednotlivé koncové body VPN jakými mohou být VPN Server a VPN Klient jsou ověřovány bezpečnými metodami, tzv. sdíleným klíčem nebo klientským certifikátem X.509 a tyto metody zabraňují přístupu neoprávněným osobám.



Obr. 6.1: Trasa virtuálního tunelu VPN mezi koncovými body spojení

6.1 Volba řešení

Pro vytvoření VPN se v současnosti nabízí několik možností. Například lze použít proprietární řešení od firem Cisco nebo Microsoft, řešení postavené na otevřeném standardu IPSEC, nebo některou z otevřených technologií, která nepoužívá IPSEC. Proprietární řešení jsou zpravidla poměrně nákladná a hodí se spíše pro větší nasazení. IPSEC je naproti tomu výrazně levnější, ale jeho konfigurace je dosti složitá a

má problém s překonáním překladu adres NAT. Do skupiny projektů, které nepoužívají IPSEC patří řešení OpenVPN. Hlavní předností tohoto řešení je multiplatformní podpora, možnost chodu v uživatelském módu, podpora režimů 1:1 (tunel) a N:1 (klient/server), možnost použít sdílený klíč nebo SSL certifikát, odolnost při použití na nekvalitních linkách, komprese a bezpečnost.

6.2 OpenVPN

OpenVPN standardně používá protokol UDP, ale lze použít i TCP. Veškerá komunikace probíhá na jednom portu a lze tedy jednoduše konfigurovat firewall s překladem adres, aby propustil tento port. U řešení PPTP je například nutné načíst speciální podporu pro firewall pro překonání NATu a nastavit pravidla pro protokoly TCP a GRE. V tomto ohledu je konfigurace OpenVPN značně jednodušší. OpenVPN démon běží v uživatelském režimu a komunikuje prostřednictvím TAP nebo TUN zařízení. Tyto rozhraní předávají veškerá přijatá data přímo uživatelskému procesu, který tak může vystupovat v roli síťové karty. Ve většině moderních linuxových distribucí je již zakompilována podpora pro TUN/TAP rozhraní. Samotná instalace OpenVPN je velice jednoduchá. Pokud chceme používat kompresi přenášených dat je potřeba mít nainstalovanou knihovnu LZO.

6.2.1 Vytvoření tunelu

Tunel v OpenVPN využívá síťové rozhraní TUN. Tato rozhraní pro účely propojení dvou bodů využívají adresy se síťovou maskou /30 a jsou v popisu rozhraní v linuxu označená jako PointToPoint, bežná ethernetová rozhraní mívají příznak Broadcast. Pro síť s maskou /30 se nastaví na rozhraní TUN první využitelná IP adresa a v dané síti je pro druhé zařízení již možná pouze jediná platná adresa. Proces vytvoření VPN síťě sestává z několika kroků. Tím prvním je vytvořit tajný klíč, kterým bude přenos mezi dvěmi síťěmi šifrován. Klíč se jednoduše vytvoří příkazem

```
openvpn --genkey --secret /etc/openvpn/secret.key
```

. Tento tajný klíč sdílí obě komunikující strany a proto je nutné klíč dopravit k protistraně bezpečným způsobem. Zde se nabízí možnost využít program SCP postavený na protokolu SSH, který přenášená data šifruje. Dalším krokem by mělo být vytvoření konfiguračních souborů a skriptů pro obě strany komunikace, kde jedna strana se konfiguruje jako server a ta druhá se konfiguruje jako klient a tímto vznikne tunel 1:1.

6.2.2 Konfigurační soubory

Klientský a serverový konfigurační soubor se liší jen minimálně a proto zde uvádím pouze konfiguraci serveru.

```
port 1194
proto udp
ifconfig 10.0.0.1 10.0.0.2
dev tun0
secret secret.key
up ./up
down ./down
```

Port 1194 je vychozí pro aplikaci OpenVPN, je možné samozřejmě použít libovný jiný. Direktiva proto nastavuje protokol UDP, který byl vybrán pro své nízké nároky při zpracování paketů a nezanáší zbytečnou režiji jako protokol TCP. Transportní vrstva postavená na protokolu UDP je vhodná i pro časově citlivé služby, například pro internetovou telefonii. Za parametrem ifconfig se specifikuje lokální IP adresa pro rozhraní tun0, druhá IP adresa patří protější straně tunelu. Parametr secret určuje způsob ověřování, v tomto případě je to metoda ověření tajným sdíleným klíčem. Na dalším řádku parametry up a down odkazují na skripty, ve kterých lze nastavit dodatečné parametry pro systém nebo spojení, zpravidla se používá pro přidávání a odebírání cest do vzdálených sítí jakmile je spojení se vzdálenou stranou navázáno případně ukončeno. Pro klientský konfigurační soubor se navíc použije direktiva remote s IP adresou nebo doménovým jménem protější strany. Spojení se navazuje ze strany klienta, server pouze naslouchá příchozím požadavkům na spojení.

Níže je zobrazen skript UP, který se vykoná po navázání spojení a přídává do směrovací tabulky trasu do vzdálené sítě.

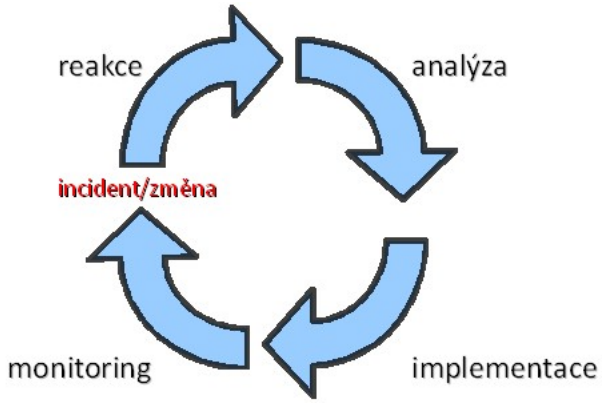
```
route add -net 172.16.100.0 netmask 255.255.255.248 gw 10.0.0.1 dev tun0
```

Pro rozsáhlejší sítě je vhodné použít některý z dynamických směrovacích protokolů, například RIP nebo OSPF. Pro systém linux existuje aplikace Quagga integrující zmíněné směrovací protokoly a mnohé další.

7 AUDIT

Ve výchozím nastavení jsou log zprávy z iptables logovány s prioritu číslo 4 - kern.warn a tyto zprávy jsou zapisovány do systémového logu ve /var/log/messages. Vytváření log zpráv je řízeno procesem syslog.

Firewall prostřednictvím zápisů do systémového logu poskytuje významné informace o své činnosti. Nutným předpokladem je, aby existovaly příslušná pravidla umožňující tento zápis a také, aby veškerá spojení procházela firewallem. Firewall s vhodným nastavením logovacích poskytuje tyto záznamy systému a odsud pak mohou být analyzovány dalšími systémy pro detekci podezřelé aktivity jakými jsou události nasvědčující probíhajícím útokům nebo skenování síťových prostředků a služeb, zpravidla se tato činnost nazýva skenování portů. Z hlediska bezpečnosti a zajištění provozuschopnosti síťe je nesmírně důležité shromažďovat síťové logy. Logy je potřeba zkoumat v pravidelných časových intervalech následně pak vyhodnocovat a zkoumat zda-li firewall obstál při pokusech o útok. Pokud se systém korektně vypořádá s těmito incidenty lze stanovit, že bezpečnostní politika firewallu je postačující. V opačném případě je žádoucí přehodnotit konfiguraci firewallu a provést patřičné změny vedoucí k nápravě. Na bezpečnost se musí pohlížet jako na dynamický proces. Bezpečnostní management je cyklus zahrnující v nejobecnějším pojetí tyto činnosti: analýzu, implementaci, monitoring a reakci. Požadavky bezpečnostní politiky jsou splněny jen tehdy provádí-li se tyto činnost neustále dokola.



Obr. 7.1: Bezpečnostní cyklus

Rozsah síťových logů je i u menších sítí značný. Ruční analýza takového množství záznamů není jednoduše proveditelná. Pro tyto účely existuje řada nástrojů. Jedním z takových nástrojů je i IDS systém PSAD, který poskytuje vhodné informace a dává dobrý přehled o bezpečnostních incidentech. Předpokladem pro funkci je existence pravidel s akcí LOG na konci definice každého řetězce firewallu případně u těch pravidel u kterých to má význam. Pro správnou funkci stačí logovat na konci

řetězců INPUT a FORWARD ze směru od veřejné sítě případně ze všech směrů. Samozřejmě je možné logovat i opačným směrem, tzn. z vnitřních sítí, avšak zde se vytrácí efektivita předpokládáme-li, že z vnitřního rozsahu bude pocházet minimální množství útočných akcí. Ve větších sítích má smysl učinit rozvahu a následně implementovat pravidla s cílem LOG jen pro určité směry.

Systém PSAD dokáže spoulupracovat se systémem FWSNORT a využít tak přeložených pravidel SNORT na ekvivalentní pravidla iptables. Tato vlastnost umožňuje zkoumat aplikační část datového paketu nad kterou je provedena kontrola proti pravidlům a posuzuje se shoda se vzorky škodlivých kódů. Pravidla zanešená do firewallu systémem FWSNORT generují log zprávy, které využívá systém PSAD pro své analýzy. Výsledkem těchto porovnávání může být následující příklad:

```
"BACKDOOR GateCrasher Connection attempt" (tcp),
Count: 2,  Unique sources: 1,  Sid: 147
```

Výše uvedený řádek specifikuje typ zkoušeného útoku, protokol, počet paketů obsahujicích dané vzorky škodlivého kódu, dále počet unikátních zdrojů, ze kterých byl útok proveden a nakonec se uvádí identifikátor pravidla SNORT, Sid. Někdy je obtížné určit zda se jedná o skutečný útok nebo jen falešný poplach. Proto je vhodné posoudit počet shod a počet unikátních zdrojů útoků.

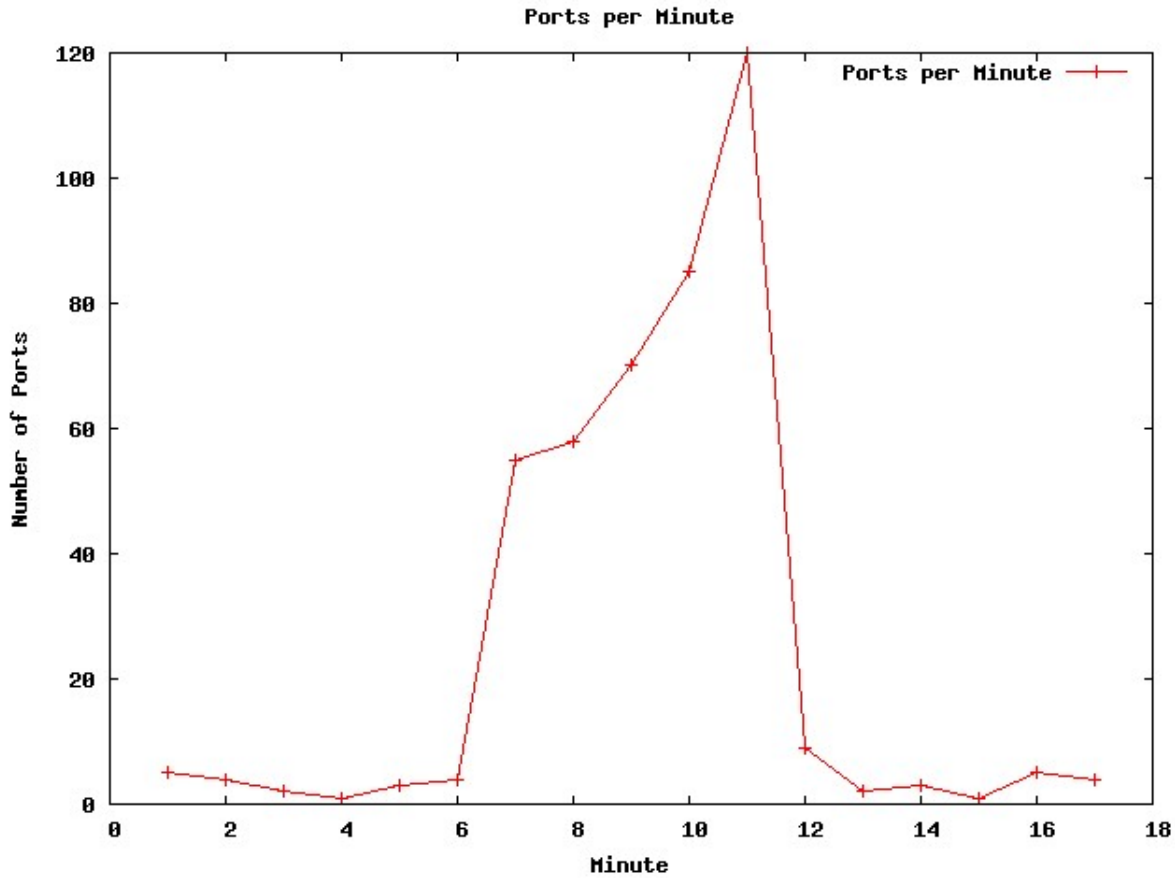
Kromě zmíněného porovnávání se vzorky nabízí PSAD i celkem přehledné statistiky v několika kategoriích. Nasledující výstup byl pro přehlednost zkrácen.

```
[+] Top 25 attackers:
      192.168.2.10    DL: 4, Packets: 3494, Sig count: 101

[+] Top 20 scanned ports:
      tcp 30666 6 packets
      tcp 22    6 packets
      tcp 44299 5 packets

[+] iptables log prefix counters:
      "retezec INPUT:": 3782
      "Neplatne pakety:": 9

    iptables auto-blocked IPs:
        [NONE]

 Total packet counters: tcp: 3388, udp: 377, icmp: 16
```

```
[+] IP Status Detail:

SRC:  192.168.2.10, DL: 4, Dsts: 3, Pkts: 3407, Unique sigs: 12,
 Email alerts: 0, Local IP
    Source OS fingerprint(s):
        SunOS:4.1::SunOS 4.1.x

    DST: 255.255.255.255, Local IP
        Scanned ports: UDP 67, Pkts: 2, Chain: INPUT, Intf: eth0
    DST: 192.168.2.15, Local IP
        Scanned ports: UDP 138, Pkts: 1, Chain: INPUT, Intf: eth0
    DST: 192.168.2.8, Local IP
        Scanned ports: TCP 1-65301, Pkts: 3388, Chain: INPUT, Intf: eth0
        Signature match: "MISC xfs overflow attempt"

   Total scan sources: 1
   Total scan destinations: 3
```

První kategorie informuje o dvacetipěti nejaktivnější zdrojích potenciálních útoků, doplněných o identifikátor DL - danger level, který označuje dosaženou úroveň nebezpečí a v závislosti na této úrovni se posuzuje míra ohrožení sítě což v konečném důsledku při dosažení úrovně DL=5 vede k zablokování zdroje útoku. Tato úroveň pak dle konfigurace PSAD vyvolává příslušné reakce - viz. kapitola 5.4.2. Ve druhé kategorii se uvádí dvacet nejčastěji zkoušených portů což poskytuje informaci o cílení útoků na konkrétní služby. Na základě poznatků z druhé kategorie lze zvážit přesunutí služeb na jiné porty a pokud to je možné pak i zakázat bannery služeb. Ve třetí kategorii jsou jednotlivé prefixi uvedené v pravidlech iptables s akcí LOG a počet shod s těmito prefixy. Ve čtvrté kategorii je podrobný přehled o zdrojích a cílech útoků. V závislosti na úrovni zaznamenaných logů a porovnáním se signaturami operačních systému lze zjistit i konkrétní operační systém ze kterého byl útok proveden.

Poslední zajímavou funkcí, kterou PSAD umí je vykreslování grafů ve spolupráci s aplikací gnuplot. Vykreslit lze téměř libovolnou veličinu ze zaznamenaných údajů. Tato vlastnost přináší lepší možnosti analýzi a poskytuje okamžitý náhled na průběh sledované veličiny v definovaném časovém rozmezí. Na obr. 7.2 je vykreslen průběh počtu zkoušených portů za jednu minutu.



Obr. 7.2: Počet zkoušených portů za jednu minutu

8 ZÁVĚR

Počítačový systém může být ohrožen různou formou vzdálených útoků. Převážná většina systémů je připojena k nějaké veřejné síti, takže je nutné se vyrovnat s určitým rizikem. Míra rizika je především dána charakterem a rozsahem poskytovaných služeb v neposlední řadě také cenou a důležitostí informací. Zmíněné aspekty mi byly podnětem při vytváření bezpečnostní politiky firewallu. Tato politika si klade za cíl definovat bezpečné formy komunikace, vypořádání se s možnými riziky a dodržení standardů pro komunikace síťových subsystémů.

Systémy IPS a IDS chrání síť před útoky jak zvenčí tak i zevnitř a zefektivňují chod celého informačního systému. Správné nastavení hraniční filtrace je klíčovým prvkem bezpečnosti celého systému.

Poskytujeme-li nějaké služby ať už dostupné z internetu nebo i z jiných sítí, je potřeba dbát na určitá bezpečnostní pravidla. Nestačí pouze takovou službu zpřístupnit, je třeba i hledět na její bezpečný provoz, vymezit správnou cestu sítí, aby v případě zdařeného útoku byl dopad na síť jako celek minimální. Mezi doplňující aspekty ovlivňující bezpečnost lze zařadit například kvalitní správu hesel, pravidelnou aplikaci záplat operačních systémů a aplikací. Řadě útoků lze předejít vypnutím nepotřebných služeb. Možnost úspěšného průniku snižuje kvalitní konfigurace síťové brány jakou je firewall. Klíčovým prvkem se pak stává i zaznamenávání bezpečnostních incidentů a jejich pravidelná analýza.

Celé řešení projektu je postavené na dílčích open-source aplikacích a systémech určených spíše pro nasazení v menších sítí jakou je i modelová síť v rámci této práce. Jednotlivé subsystémy jsou implementovány za účelem maximální efektivity chodu sítě se vzájemnou kooperací při dodržení bezpečnostních standardů. Řešení je tedy vhodné použít i jako základ pro podnikové sítě.

Úměrně s vývojem internet rostou i hrozby, které denně představují riziko pro informační systémy. Sofistikovanost útoků také neustále narůstá. Bohužel obranné prvky jsou v tomto ohledu vždy o nějaký ten krůček pozadu a stále je prostor ke zlepšování současných technologií a postupů implementace.

LITERATURA

[1] Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman
Oreilly - Building Internet Firewalls
ISBN: 1-56592-871-7

[2] Olaf Kirch & Terry Dawson
Linux Network Administrator's Guide
ISBN: 1-56592-400-2

[3] Chris Hare & Karanjit Siyan
Internet Firewalls and Network Security
ISBN: 1-56205-508-

[4] Rusty Russell
Linux 2.4 Packet Filtering HOWTO
Dostupné z URL:
<http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html>.

[5] Michael Rash
psad: Intrusion Detection and Log Analysis with iptables
Dostupné z URL:
<http://www.cipherdyne.org/psad/docs/>.

[6] Michael Rash
fwsnort: Application Layer IDS/IPS with iptables
Dostupné z URL:
<http://www.cipherdyne.org/fwsnort/docs/>.

[7] Fred Baker
Requirements for IP Version 4 Routers
Dostupné z URL:
<http://www.ietf.org/rfc/rfc1812.txt>.

SEZNAM PŘÍLOH

A PŘÍLOHA A

A.1 skript FIREWALL VPN GATEWAY

```
#!/bin/sh

### Nastaveni promennych ###
IPT="/sbin/iptables"
MOD="/sbin/modprobe"

INET_IP="192.168.2.8"
INET_IFACE="eth0"
INET_BROADCAST="192.168.2.15"

LAN_IP="172.16.250.1"
LAN_IP_RANGE="172.16.250.0/29"
LAN_IFACE="eth1"

DMZ_IP="172.16.200.1"
DMZ_IP_RANGE="172.16.200.0/29"
DMZ_IFACE="eth2"

VPN_IP="10.0.0.1"
VPN_IP_RANGE="10.0.0.0/30"
VPN_IFACE="tun0"

LOCAL_IP_RANGE="172.16.0.0/16"

LO_IFACE="lo"
LO_IP="127.0.0.1"

### ---------------------------- ###

### Parametry skriptu pro vypnuti a reset FW ###
if [ "$1" == "stop" ]; then
$IPT --flush
$IPT -t nat --flush
$IPT -t mangle --flush
$IPT -X
```

```
$IPT -t nat -X
$IPT -t mangle -X
$IPT -P OUTPUT  ACCEPT
$IPT -P INPUT   ACCEPT
$IPT -P FORWARD ACCEPT

echo "########################################"
echo "# Firewall zastaven, politika ACCEPT ! #"
echo "########################################"

   exit 0;
fi

if [ "$1" == "restart" ]; then
$IPT --flush
$IPT -t nat --flush
$IPT -t mangle --flush
  $IPT -X
  $IPT -t nat -X
  $IPT -t mangle -X

echo "########################"
echo "# Firewall resetovan ! #"
echo "########################"

  /etc/init.d/firewall.sh
  exit 0;
fi

### Nacteni potrebnych modulu ###
/sbin/depmod -a

$MOD ip_tables
$MOD ip_conntrack
$MOD iptable_filter
$MOD iptable_mangle
$MOD iptable_nat
$MOD ipt_LOG
$MOD ipt_limit
```

```
$MOD ipt_state
$MOD ip_gre
$MOD ip_conntrack_pptp
$MOD ip_nat_pptp

### Inicializace bezpecnostniho nastaveni ###

### Povoleni smerovani ###
echo 1 > /proc/sys/net/ipv4/ip_forward

### Omezeni echo broadcast ###
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

### Omezeni zdrojoveho smerovani ###
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
echo 0 > $f
done

### Zapnuti TCP SYN cookies ###
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

### Aktivace anti-spoofing ###
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
echo 1 > $f
done

### Vychozi politika - DROP ###
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP
$IPT -t nat --policy PREROUTING DROP
$IPT -t nat --policy OUTPUT DROP
$IPT -t nat --policy POSTROUTING DROP
$IPT -t mangle --policy PREROUTING DROP
$IPT -t mangle --policy OUTPUT DROP

### Logovani neplatnych paketu na INPUT a FORWARD ###
$IPT -N logdrop
$IPT -A INPUT -m state --state INVALID -j logdrop
```

```
$IPT -A FORWARD -m state --state INVALID -j logdrop
$IPT -A logdrop -m limit --limit 2/s --limit-burst 3 -j LOG \
--log-prefix "Neplatne pakety: "
$IPT -A logdrop -j DROP

### Povoleni sluzeb smerujicich na firewall z vnejsich siti ###
### Pravidla definuji pouze pristup k "bezpecnym" sluzbam ###
$IPT -A INPUT -i $INET_IFACE -p tcp --dport ssh -m state \
--state NEW -m recent --set --name SSH -j ACCEPT
$IPT -A INPUT -i $INET_IFACE -p tcp --dport ssh -m recent \
--update --seconds 60 --hitcount 4 --rttl --name SSH -j LOG \
--log-prefix "SSH bruteforce: "
$IPT -A INPUT -i $INET_IFACE -p tcp --dport ssh -m recent \
--update --seconds 60 --hitcount 4 --rttl --name SSH -j DROP

$IPT -A INPUT -i $INET_IFACE -p udp --dport 1194 -j ACCEPT

### Odmitnuti sluzby AUTH ###
$IPT -A INPUT -i $INET_IFACE -p tcp --dport auth \
-j REJECT --reject-with tcp-reset

### Nastaveni povolenych ICMP paketu na firewall ###
### Echo reply, Destination unreachable, Echo request, Time exceeded ###
$IPT -N ICMP_pakety
$IPT -A INPUT -i $INET_IFACE -p icmp -j ICMP_pakety
$IPT -A ICMP_pakety -p icmp --icmp-type 3 -m length -j ACCEPT
$IPT -A ICMP_pakety -p icmp --icmp-type 8 -m length \
--length 28:92 -m limit --limit 2/s --limit-burst 5 -j ACCEPT
$IPT -A ICMP_pakety -p icmp --icmp-type 11 -j ACCEPT

### Vstupni pravidla ###
### Pakety na vstupy DMZ a INET pouze z vnitrne navazanych spojeni ###
$IPT -A INPUT -p ALL -i $LAN_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -i $VPN_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -i $LO_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -s ! $LOCAL_IP_RANGE -i $DMZ_IFACE \
-m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A INPUT -p ALL -s ! $LOCAL_IP_RANGE -i $INET_IFACE \
-m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
### Povoleni odchoziho provozu ###
$IPT -A OUTPUT -s $LO_IP -p ALL -j ACCEPT
$IPT -A OUTPUT -o $LAN_IFACE -j ACCEPT
$IPT -A OUTPUT -o $VPN_IFACE -j ACCEPT
$IPT -A OUTPUT -p icmp -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p udp --dport 53 -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --dport www -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p udp --dport https -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --dport https -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p udp --sport 1194 -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --sport ssh -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --dport whois -j ACCEPT

### Pravidla smerovani a povolene stavy spojeni ###
$IPT -A FORWARD -i $LAN_IFACE -j ACCEPT
$IPT -A FORWARD -i $VPN_IFACE -j ACCEPT
$IPT -A FORWARD -i $DMZ_IFACE -o $INET_IFACE -j ACCEPT
$IPT -A FORWARD -i $DMZ_IFACE -o $LAN_IFACE -m state \
--state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $INET_IFACE -o $LAN_IFACE -m state \
--state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $INET_IFACE -o $DMZ_IFACE -m state \
--state ESTABLISHED,RELATED -j ACCEPT

### Povolene sluzby smerujici do DMZ ###
$IPT -N dmz_forward
$IPT -A FORWARD -i $INET_IFACE -o $DMZ_IFACE -m state \
--state NEW -j dmz_forward
$IPT -A dmz_forward -p tcp --dport www -j ACCEPT
$IPT -A dmz_forward -p tcp --dport https -j ACCEPT
$IPT -A dmz_forward -p udp --dport https -j ACCEPT
$IPT -A dmz_forward -p gre -j ACCEPT
$IPT -A dmz_forward -p tcp --dport 1723 -j ACCEPT

### Preklad zdrojovych adres pro pristup k vnejsi siti ###
### Nebo -j MASQUERADE v pripade dynamicke IP adresa vnejsiho rozhrani ###
$IPT -t nat -A POSTROUTING -o $INET_IFACE -j SNAT \
--to-source $INET_IP
```

```
### Mapovani portu a protokolu do DMZ ###
$IPT -t nat -N dmz_prerouting
$IPT -t nat -A PREROUTING -i $INET_IFACE -j dmz_prerouting
$IPT -t nat -A dmz_prerouting -p tcp --dport www -j DNAT \
--to 172.16.200.2
$IPT -t nat -A dmz_prerouting -p tcp --dport https -j DNAT \
--to 172.16.200.2
$IPT -t nat -A dmz_prerouting -p udp --dport https -j DNAT \
--to 172.16.200.2
$IPT -t nat -A dmz_prerouting -p tcp --dport 1723 -j DNAT \
--to 172.16.200.2
$IPT -t nat -A dmz_prerouting -p gre -j DNAT --to 172.16.200.2

### Nastaveni vystupni hodnoty TTL ###
### Pouze pro zmateni nepritele ###
$IPT -t mangle -A POSTROUTING -o $INET_IFACE -j TTL --ttl-set 70

echo "########################"
echo "# Firewall nastaven ! #"
echo "########################"

$IPT -A INPUT -j LOG --log-prefix "retezec INPUT: " --log-tcp-options \
--log-tcp-sequence --log-ip-options
$IPT -A FORWARD -j LOG --log-prefix "retezec FORWARD: " --log-tcp-options \
--log-tcp-sequence --log-ip-options
echo "###################################"
echo "# Podpora pro IDS PSAD zapnuta ! #"
echo "###################################"

fwsnort --strict --ipt-reject --ipt-apply
echo "############################"
echo "# IPS system je aktivni ! #"
echo "############################"
```

B PŘÍLOHA B

B.1 skript FIREWALL VPN

```
#!/bin/sh

### Nastaveni promennych ###
IPT="/sbin/iptables"
MOD="/sbin/modprobe"

INET_IP="192.168.2.5"
INET_IFACE="eth4"
INET_BROADCAST="192.168.2.15"

LAN_IP="172.16.100.1"
LAN_IP_RANGE="172.16.100.0/29"
LAN_IFACE="eth5"

VPN_IP="10.0.0.2"
VPN_IP_RANGE="10.0.0.0/30"
VPN_IFACE="tun0"

LOCAL_IP_RANGE="172.16.0.0/16"

LO_IFACE="lo"
LO_IP="127.0.0.1"

### ------------------------------ ###

### Parametry skriptu pro vypnuti a reset FW ###
if [ "$1" == "stop" ]; then
$IPT --flush
$IPT -t nat --flush
$IPT -t mangle --flush
$IPT -X
$IPT -t nat -X
$IPT -t mangle -X
$IPT -P OUTPUT  ACCEPT
$IPT -P INPUT   ACCEPT
```

```
$IPT -P FORWARD ACCEPT

echo "########################################"
echo "# Firewall zastaven, politika ACCEPT ! #"
echo "########################################"

   exit 0;
fi

if [ "$1" == "restart" ]; then
$IPT --flush
$IPT -t nat --flush
$IPT -t mangle --flush
  $IPT -X
  $IPT -t nat -X
  $IPT -t mangle -X

echo "########################"
echo "# Firewall resetovan ! #"
echo "########################"

  /etc/init.d/firewall-vpn.sh
  exit 0;
fi

### Nacteni potrebnych modulu ###
/sbin/depmod -a

$MOD ip_tables
$MOD ip_conntrack
$MOD iptable_filter
$MOD iptable_mangle
$MOD iptable_nat
$MOD ipt_LOG
$MOD ipt_limit
$MOD ipt_state

### Inicializace bezpecnostniho nastaveni ###
```

```
### Povoleni smerovani ###
echo 1 > /proc/sys/net/ipv4/ip_forward

### Omezeni echo broadcast ###
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

### Omezeni zdrojoveho smerovani ###
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
echo 0 > $f
done

### Zapnuti TCP SYN cookies ###
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

### Aktivace anti-spoofing ###
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
echo 1 > $f
done

### Vychozi politika - DROP ###
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP
$IPT -t nat --policy PREROUTING DROP
$IPT -t nat --policy OUTPUT DROP
$IPT -t nat --policy POSTROUTING DROP
$IPT -t mangle --policy PREROUTING DROP
$IPT -t mangle --policy OUTPUT DROP

### Logovani neplatnych paketu na INPUT a FORWARD ###
$IPT -N logdrop
$IPT -A INPUT -m state --state INVALID -j logdrop
$IPT -A FORWARD -m state --state INVALID -j logdrop
$IPT -A logdrop -m limit --limit 2/s --limit-burst 3 -j LOG \
--log-prefix "Neplatne pakety: "
$IPT -A logdrop -j DROP

### Povoleni sluzeb smerujicich na firewall z vnejsich siti ###
### Pravidla definuji pouze pristup k "bezpecnym" sluzbam ###
```

```
$IPT -A INPUT -i $INET_IFACE -p tcp --dport ssh -m state \
--state NEW -m recent --set --name SSH -j ACCEPT
$IPT -A INPUT -i $INET_IFACE -p tcp --dport ssh -m recent \
--update --seconds 60 --hitcount 4 --rttl --name SSH -j LOG \
--log-prefix "SSH bruteforce: "
$IPT -A INPUT -i $INET_IFACE -p tcp --dport ssh -m recent \
--update --seconds 60 --hitcount 4 --rttl --name SSH -j DROP

$IPT -A INPUT -i $INET_IFACE -p udp --dport 1194 -j ACCEPT

### Odmitnuti sluzby AUTH ###
$IPT -A INPUT -i $INET_IFACE -p tcp --dport auth \
-j REJECT --reject-with tcp-reset

### Nastaveni povolenych ICMP paketu na firewall ###
### Echo reply, Destination unreachable, Echo request, Time exceeded ###
$IPT -N ICMP_pakety
$IPT -A INPUT -i $INET_IFACE -p icmp -j ICMP_pakety
$IPT -A ICMP_pakety -p icmp --icmp-type 8 -m length \
--length 28:92 -m limit --limit 2/s --limit-burst 5 -j ACCEPT

### Vstupni pravidla ###
### Pakety na vstupy DMZ a INET pouze z vnitrne navazanych spojeni ###
$IPT -A INPUT -p ALL -i $LAN_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -i $VPN_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -i $LO_IFACE -j ACCEPT
$IPT -A INPUT -p ALL -s ! $LOCAL_IP_RANGE -i $INET_IFACE \
-m state --state ESTABLISHED,RELATED -j ACCEPT

### Povoleni odchoziho provozu ###
$IPT -A OUTPUT -s $LO_IP -p ALL -j ACCEPT
$IPT -A OUTPUT -o $LAN_IFACE -j ACCEPT
$IPT -A OUTPUT -o $VPN_IFACE -j ACCEPT
$IPT -A OUTPUT -p icmp -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p udp --dport 53 -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --dport www -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p udp --sport 1194 -j ACCEPT
$IPT -A OUTPUT -o $INET_IFACE -p tcp --sport ssh -j ACCEPT
```

```
### Pravidla smerovani a povolene stavy spojeni ###
$IPT -A FORWARD -i $LAN_IFACE -j ACCEPT
$IPT -A FORWARD -i $VPN_IFACE -j ACCEPT
$IPT -A FORWARD -i $INET_IFACE -o $LAN_IFACE -m state \
--state ESTABLISHED,RELATED -j ACCEPT

### Preklad zdrojovych adres pro pristup k vnejsi siti ###
### Nebo -j MASQUERADE v pripade dynamicke IP adresa vnejsiho rozhrani ###
$IPT -t nat -A POSTROUTING -o $INET_IFACE -j SNAT \
--to-source $INET_IP

echo "########################"
echo "# Firewall nastaven ! #"
echo "########################"
```

C PŘÍLOHA C

C.1 konfigurační soubor psad.conf

```
EMAIL_ADDRESSES             root@localhost;
HOSTNAME                    debian;
HOME_NET                    172.16.0.0/16;
EXTERNAL_NET                any;
FW_SEARCH_ALL               Y;
#FW_MSG_SEARCH               Audit;
#FW_MSG_SEARCH               REJECT;
SYSLOG_DAEMON               syslogd;
DANGER_LEVEL1               5;    ### Number of packets.
DANGER_LEVEL2               15;
DANGER_LEVEL3               80;
DANGER_LEVEL4               500;
DANGER_LEVEL5               4000;
CHECK_INTERVAL              5;
SNORT_SID_STR               SID;
PORT_RANGE_SCAN_THRESHOLD   10;
ENABLE_PERSISTENCE          N;
SCAN_TIMEOUT                3600;  ### seconds
SHOW_ALL_SIGNATURES         N;
ALERTING_METHODS            noemail;
ENABLE_SYSLOG_FILE          N;
IPT_WRITE_FWDATA            N;
IPT_SYSLOG_FILE             /var/log/messages;
ENABLE_SIG_MSG_SYSLOG       Y;
SIG_MSG_SYSLOG_THRESHOLD    10;
SIG_SID_SYSLOG_THRESHOLD    10;
MAX_HOPS                    20;
IGNORE_KERNEL_TIMESTAMP     Y;
IGNORE_CONNTRACK_BUG_PKTS   Y;
IGNORE_PORTS                NONE;
IGNORE_PROTOCOLS            NONE;
IGNORE_INTERFACES           NONE;
IGNORE_LOG_PREFIXES         NONE;
MIN_DANGER_LEVEL            1;
```

```
EMAIL_ALERT_DANGER_LEVEL    1;
ENABLE_INTF_LOCAL_NETS      Y;
ENABLE_MAC_ADDR_REPORTING   N;
ENABLE_FW_LOGGING_CHECK     Y;
EMAIL_LIMIT                 0;
ENABLE_EMAIL_LIMIT_PER_DST  N;
EMAIL_LIMIT_STATUS_MSG      Y;
ALERT_ALL                   Y;
IMPORT_OLD_SCANS            N;
SYSLOG_IDENTITY             psad;
SYSLOG_FACILITY             LOG_LOCAL7;
SYSLOG_PRIORITY             LOG_INFO;
TOP_PORTS_LOG_THRESHOLD     500;
STATUS_PORTS_THRESHOLD      20;
TOP_SIGS_LOG_THRESHOLD      500;
STATUS_SIGS_THRESHOLD       50;
TOP_IP_LOG_THRESHOLD        500;
STATUS_IP_THRESHOLD         25;
TOP_SCANS_CTR_THRESHOLD     1;
ENABLE_DSHIELD_ALERTS       N;
DSHIELD_ALERT_EMAIL         reports@dshield.org;
DSHIELD_ALERT_INTERVAL      6;  ### hours
DSHIELD_USER_ID             0;
DSHIELD_USER_EMAIL          NONE;
DSHIELD_DL_THRESHOLD        0;
HTTP_SERVERS                $HOME_NET;
SMTP_SERVERS                $HOME_NET;
DNS_SERVERS                 $HOME_NET;
SQL_SERVERS                 $HOME_NET;
TELNET_SERVERS              $HOME_NET;
AIM_SERVERS                 [64.12.24.0/24, 64.12.25.0/24, 205.188.9.0/24];
HTTP_PORTS                  80;
SHELLCODE_PORTS             !80;
ORACLE_PORTS                1521;
ENABLE_SNORT_SIG_STRICT     Y;
ENABLE_AUTO_IDS             Y;
AUTO_IDS_DANGER_LEVEL       5;
AUTO_BLOCK_TIMEOUT          3600;
ENABLE_AUTO_IDS_REGEX       N;
```

```
AUTO_BLOCK_REGEX            ESTAB;  ### from fwsnort logging prefixes
ENABLE_RENEW_BLOCK_EMAILS   N;
ENABLE_AUTO_IDS_EMAILS      N;
IPTABLES_BLOCK_METHOD       Y;
IPT_AUTO_CHAIN1             DROP, src, filter, INPUT, 1,
 PSAD_BLOCK_INPUT, 1;
IPT_AUTO_CHAIN2             DROP, dst, filter, OUTPUT, 1,
 PSAD_BLOCK_OUTPUT, 1;
IPT_AUTO_CHAIN3             DROP, both, filter, FORWARD, 1,
 PSAD_BLOCK_FORWARD, 1;
FLUSH_IPT_AT_INIT           Y;
IPTABLES_PREREQ_CHECK       1;
TCPWRAPPERS_BLOCK_METHOD    N;
WHOIS_TIMEOUT               60;  ### seconds
WHOIS_LOOKUP_THRESHOLD      20;
DNS_LOOKUP_THRESHOLD        20;
ENABLE_EXT_SCRIPT_EXEC      N;
EXTERNAL_SCRIPT             /bin/true;
EXEC_EXT_SCRIPT_PER_ALERT   N;
DISK_CHECK_INTERVAL         300;  ### seconds
DISK_MAX_PERCENTAGE         95;
DISK_MAX_RM_RETRIES         10;
ENABLE_SCAN_ARCHIVE         N;
TRUNCATE_FWDATA             Y;
MIN_ARCHIVE_DANGER_LEVEL    4;
MAIL_ALERT_PREFIX           [psad-alert];
MAIL_STATUS_PREFIX          [psad-status];
MAIL_ERROR_PREFIX           [psad-error];
MAIL_FATAL_PREFIX           [psad-fatal];
SIG_UPDATE_URL              http://www.cipherdyne.org/psad/signatures;
PSADWATCHD_CHECK_INTERVAL   5;  ### seconds
PSADWATCHD_MAX_RETRIES      10;

### Directories
PSAD_DIR                    /var/log/psad;
PSAD_RUN_DIR                /var/run/psad;
PSAD_FIFO_DIR               /var/lib/psad;
PSAD_LIBS_DIR               /usr/lib/psad;
PSAD_CONF_DIR               /etc/psad;
```

```
PSAD_ERR_DIR                $PSAD_DIR/errs;
CONF_ARCHIVE_DIR            $PSAD_CONF_DIR/archive;
SCAN_DATA_ARCHIVE_DIR       $PSAD_DIR/scan_archive;
ANALYSIS_MODE_DIR           $PSAD_DIR/ipt_analysis;
SNORT_RULES_DIR             $PSAD_CONF_DIR/snort_rules;
FW_DATA_FILE                $PSAD_DIR/fwdata;
ULOG_DATA_FILE              $PSAD_DIR/ulogd.log;
FW_CHECK_FILE               $PSAD_DIR/fw_check;
DSHIELD_EMAIL_FILE          $PSAD_DIR/dshield.email;
SIGS_FILE                   $PSAD_CONF_DIR/signatures;
ICMP_TYPES_FILE             $PSAD_CONF_DIR/icmp_types;
AUTO_DL_FILE                $PSAD_CONF_DIR/auto_dl;
SNORT_RULE_DL_FILE          $PSAD_CONF_DIR/snort_rule_dl;
POSF_FILE                   $PSAD_CONF_DIR/posf;
POF_FILE                    $PSAD_CONF_DIR/pf.os;
IP_OPTS_FILE                $PSAD_CONF_DIR/ip_options;
PSAD_FIFO_FILE              $PSAD_FIFO_DIR/psadfifo;
ETC_HOSTS_DENY_FILE         /etc/hosts.deny;
ETC_SYSLOG_CONF             /etc/syslog.conf;
ETC_RSYSLOG_CONF            /etc/rsyslog.conf;
ETC_SYSLOGNG_CONF           /etc/syslog-ng/syslog-ng.conf;
ETC_METALOG_CONF            /etc/metalog/metalog.conf;
STATUS_OUTPUT_FILE          $PSAD_DIR/status.out;
ANALYSIS_OUTPUT_FILE        $PSAD_DIR/analysis.out;
INSTALL_LOG_FILE            $PSAD_DIR/install.log;
PSAD_PID_FILE               $PSAD_RUN_DIR/psad.pid;
PSAD_CMDLINE_FILE           $PSAD_RUN_DIR/psad.cmd;
KMSGSD_PID_FILE             $PSAD_RUN_DIR/kmsgsd.pid;
PSADWATCHD_PID_FILE         $PSAD_RUN_DIR/psadwatchd.pid;
AUTO_BLOCK_IPT_FILE         $PSAD_DIR/auto_blocked_iptables;
AUTO_BLOCK_TCPWR_FILE       $PSAD_DIR/auto_blocked_tcpwr;
AUTO_IPT_SOCK               $PSAD_RUN_DIR/auto_ipt.sock;
FW_ERROR_LOG                $PSAD_ERR_DIR/fwerrorlog;
PRINT_SCAN_HASH             $PSAD_DIR/scan_hash;
PROC_FORWARD_FILE           /proc/sys/net/ipv4/ip_forward;
PACKET_COUNTER_FILE         $PSAD_DIR/packet_ctr;
TOP_SCANNED_PORTS_FILE      $PSAD_DIR/top_ports;
TOP_SIGS_FILE               $PSAD_DIR/top_sigs;
TOP_ATTACKERS_FILE          $PSAD_DIR/top_attackers;
```

```
DSHIELD_COUNTER_FILE        $PSAD_DIR/dshield_ctr;
IPT_PREFIX_COUNTER_FILE     $PSAD_DIR/ipt_prefix_ctr;
IPT_OUTPUT_FILE             $PSAD_DIR/psad.iptout;
IPT_ERROR_FILE              $PSAD_DIR/psad.ipterr;

iptablesCmd      /sbin/iptables;
shCmd            /bin/sh;
wgetCmd          /usr/bin/wget;
gzipCmd          /bin/gzip;
mknodCmd         /bin/mknod;
psCmd            /bin/ps;
mailCmd          /usr/bin/mail;
sendmailCmd      /usr/sbin/sendmail;
ifconfigCmd      /sbin/ifconfig;
killallCmd       /sbin/killall;
netstatCmd       /bin/netstat;
unameCmd         /bin/uname;
whoisCmd         /usr/bin/whois_psad;
dfCmd            /bin/df;
fwcheck_psadCmd  /usr/sbin/fwcheck_psad;
psadwatchdCmd    /usr/sbin/psadwatchd;
kmsgsdCmd        /usr/sbin/kmsgsd;
psadCmd          /usr/sbin/psad;
```