

Server, router a WiFi AP pro přístup studentů a hostů školy do sítě

Jakub Sporek

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Univerzita Tomáše Bati ve Zlíně	Studijní program: Inženýrská informatika
Fakulta technologická	Akademický rok: 2005/2006

PODKLAD PRO ZADÁNÍ BAKALÁŘSKÉ PRÁCE

PŘEDKLÁDÁ:	ADRESA:	OSOBNÍ ČÍSLO:
SPOREK Jakub	Kociánov 34 ; Loučná nad Desnou	T023649

PRACOVNÍ NÁZEV TÉMATU:

Server, router a WiFi AP pro přístup studentů a hostů školy do sítě

VEDOUCÍ PRÁCE:

Ing. Tomáš Dulík

ZÁSADY PRO VYPRACOVÁNÍ:

Nakonfigurujte server s OS Linux tak, aby ho bylo možno použít jako router pro bezdrátový (WiFi) přístup studentů a hostů školy do sítě a internetu. Server musí mít následující funkce:

- Routování z bezdrátové sítě do školní sítě a internetu
- Radius server pro autentizaci jednotlivých uživatelů sítě
- Překlad N neveřejných IP adres bezdrátové sítě na jednu veřejnou IP adresu školní sítě (tzv. Masquerade)
- Firewall pro uzavření všech nepotřebných portů sítě
- DHCP server pro dynamické přidělování IP adres z neveřejného rozsahu
- Spravedlivé přidělování kapacity linky (tzn. zajištění tzv. Quality Of Service ? QoS ? pomocí nějakého algoritmu Fair User Policy ? např. HTB nebo WRR) jednotlivým připojeným počítačům tak, aby uživatel nemohl získat veškerou kapacitu linky do internetu otevřením mnoha TCP spojení
- Transparentní HTTP proxy Squid, urychlující přístup k často navštěvovaným stránkám
- HTTP rozhraní s funkcemi administrace (např. zákaz přístupu do sítě z MAC adres narušitelů) a se statistikami přenesených dat jednotlivých uživatelů

SEZNAM DOPORUČENÉ LITERATURY:

- Linux Advanced Routing & Traffic Kontrol ? <http://lartc.org/>
- Radius server - <http://www.gnu.org/software/radius/radius.html>
- Poptop - open source VPN server - <http://www.poptop.org/>
- Shorewall ? firewall for IP Tables - <http://shorewall.net>
- Squid proxy HTTP cache - <http://www.squid-cache.org/>

PODPIS VEDOUCÍHO PRÁCE: _____ **DATUM:** _____

ABSTRAKT

RESUMÉ

Práce popisuje konfiguraci linuxového serveru jako routeru a AP pro WiFi připojení. Je zahrnut popis konfigurace všech potřebných programů, jako je FreeRADIUS, Shorewall, Squid, PPTPD. Server je založen na distribuci Debian GNU/Linux.

SUMMARY

The work describes configuration of the linux server as an router and AP for WiFi connection. Configuration of all needed programs such as FreeRADIUS, Shorewall, Squid and PPTPD is included. The server is based on Debian GNU/Linux distribution.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu mé bakalářské práce, a to panu Ing. Tomáši Dulíkovi za cenné rady a připomínky při konfiguraci serveru.

Well, let's just say, 'if your VCR is still blinking 12:00, you don't want Linux'.

- neznámý

OBSAH

PODĚKOVÁNÍ.....	4
ÚVOD.....	6
I. TEORETICKÁ ČÁST.....	7
1 FIREWALL V OS LINUX.....	8
1.1 IPTABLES.....	9
1.1.1 Shorewall.....	10
2 DHCP SERVER.....	11
3 CACHING PROXY SERVER.....	13
3.1 SQUID.....	13
4 AUTENTIZAČNÍ SERVER RADIUS.....	15
4.1 FREERADIUS.....	15
5 VIRTUÁLNÍ PRIVÁTNÍ SÍŤ (VPN).....	16
5.1 PPTPD (PoPToP).....	16
6 QUALITY OF SERVICE (QOS).....	17
6.1 HTB.....	19
II. PRAKTICKÁ ČÁST.....	21
7 VÝBĚR VHODNÉHO SYSTÉMU.....	22
7.1 m0N0WALL.....	23
7.2 STAROS.....	24
7.3 DEBIAN-CZF.....	27
8 KONFIGURACE SERVERU (DEBIAN GNU/LINUX).....	28
8.1 FIREWALL & MASQUERADE.....	28
8.2 SQUID.....	30
8.3 DHCP SERVER.....	31
8.4 VPN.....	31
8.4.1 PPTPD.....	32
8.4.2 FreeRADIUS.....	33
8.5 HTB – PROMETHEUS QoS.....	33
ZÁVĚR.....	35
SEZNAM POUŽITÉ LITERATURY.....	36
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	37
SEZNAM OBRÁZKŮ.....	38
SEZNAM TABULEK.....	39

ÚVOD

Internet je v dnešní době běžnou součástí života. Více lidí než kdy dříve ho používá stále častěji a častěji a internetoví provideři se předbíhají s lepšími a rychlejšími službami. Nabízejí internet po kabelové televizi, přes analogové i digitální modemy, začíná se testovat internet přes elektrickou síť a také bezdrátově přes satelit, mikrovlny nebo WiFi.

Posledně jmenované, bezdrátové sítě, v poslední době zažívají čím dál větší rozmach. Jejich instalace je na jednu stranu jednodušší na výstavbu a technickou realizaci, protože není třeba pokládat žádnou kabeláž, na druhou stranu nabízejí podstatně nižší rychlosti než nejmodernější ethernetové kabelové sítě. Tyto rychlosti jsou ale plně dostačující například pro poskytování internetu, takže jejich výhody mohou snadno převážet. Ostatně internetových providerů, poskytujících WiFi připojení, neustále přibývá.

Hlavním podnětem pro vytvoření této práce byla potřeba serveru, který by poskytoval internet uživatelům, kteří vlastní například notebook s WiFi kartou a chtějí mít možnost se připojovat kdekoliv, nebo jsou v dosahu signálu a chtějí z pohodlí domova surfovat na rychlém internetu.

Při výstavbě bezdrátové sítě a konfiguraci serverů, které mají poskytovat internetové připojení pomocí WiFi je ovšem nutno zohlednit minimálně dvě hlavní úskalí, a to bezpečnost a soukromí uživatelů. Tyto dvě položky jsou velice důležité, a to proto, že zvláště v případě bezdrátových sítí je relativně jednoduché odposlechnout jakýkoliv přenos a zjistit tak uživatelská jména i s hesly. Z tohoto důvodu je nutné veškerý přenos šifrovat a tím případným útočníkům výrazně snížit šance na úspěch.

Dále je nutno zabránit nežádoucím uživatelům přístupu do sítě vůbec. Toho se většinou dosahuje některým z autentizačních serverů, jako například RADIUS. V mé práci je to server, který má název FreeRadius. Ten komunikuje s uživatelem šifrovaně a tudíž chrání jeho soukromí před únikem důležitých informací.

I. TEORETICKÁ ČÁST

1 FIREWALL V OS LINUX

Firewally obecně jsou systémy sídlící na hranici mezi několika sítěmi, které zpracovávají provoz jdoucí přes tuto hranici (z obou směrů). Firewally umožňují filtraci a monitorování provozu. Pokročilejší implementace do síťového provozu i aktivně zasahují. Nejčastější nasazení firewallu bývá mezi LAN a Internetem, kdy chceme počítače v lokální síti chránit před útoky z Internetu.

Veškerá komunikace síťových služeb probíhá skrze porty. A právě přes tyto porty lze vést útok proti PC nebo celé síti. Stačí naskenovat porty a služby, které na nich běží a pokud obsahují bezpečnostní díru, lze systém infiltrovat. Postupů jak proniknout do sítě je samozřejmě více.

Seznam základních portů:

- FTP: 20, 21
- SSH: 22
- HTTP: 80
- POP3: 110
- IMAP3: 220

Firewall má za úkol kontrolovat všechny porty a komunikaci na nich probíhající. Tu vyhodnocuje podle pravidel, jež má nastaveny a rozhoduje, zda je komunikace vyhovující či nevyhovující. Pokud narazí na nějaké pokusy o proniknutí, pokouší se je odrazit a nepustit je dovnitř počítače.

Rozlišujeme dva základní typy firewallů:

- **proxy, aplikační brány**

Aplikační proxy pracují na úrovni protokolů jednotlivých služeb.

- **paketové filtry**

Paketové filtry sledují pohyb dat po síti po jednotlivých paketech. Podle daných pravidel posuzují jednotlivé pakety a rozhodují o jejich dalším osudu. Minimálně však určují zda se paket propustí nebo zahodí. Paketové filtry nerozumějí vyšším vrstvám než síťové a nedokáží se podle nich rozhodovat. To co mají k dispozici jsou atributy paketů jako zdrojová/cílová adresa, rozhraní, ze kterého paket přišel,

zdrojový/cílový port, etc. Samotný paketový filtr je rychlý, nenáročný na systémové zdroje. Celý proces se odehrává v jádře.

V linuxovém jádře je vestavěna poměrně silná a flexibilní podpora pro filtrování paketů (speciálně IP datagramů, i když je možno filtrovat i jiné protokoly, např. IPX) zpracovávaných síťovým subsystémem jádra. Filtrování je možno provádět ve třech různých fázích: bezprostředně po přijetí datagramu ze sítě, během jeho přesměrování do jiné sítě a před vysláním datagramu do sítě.

Na začátku se snaží jádro rozhodnout, zda-li je příchozí paket určen pro tento počítač, nebo jestli je potřeba jej routovat jinam.

- Je-li adresátem on sám, předá paket k dalšímu zpracování do vstupního (INPUT) řetězce. Pokud vyhoví tamějším filtrovacím pravidlům, dostane jej ke zpracování některý z lokálních programů poslouchající na cílovém portu.
- Pokud je datagram určen někomu jinému a počítač je zkonfigurován jako router, tedy pokud je povoleno routování paketů proměnnou `/proc/sys/net/ipv4/ip_forward == 1`, paket bude postoupen do řetězce FORWARD a počítač se jej pak pokusí podle svých možností doručit příjemci. Pokud je směrování paketů zakázáno (implicitní stav), bude paket zahozen.
- Poslední možností je, že datagram vytvořil některý z lokálních programů. Potom je nutné, aby paketový filtr opustil přes řetězec OUTPUT.

Kromě zmíněných řetězců INPUT, OUTPUT a FORWARD existují ještě další dva, a to PREROUTING a POSTROUTING, které se ovšem používají především k jiným účelům než k filtrování paketů. [8]

1.1 Iptables

Nyní k vlastnímu filtru paketů. Ten je implementován sadou modulů jádra, které se zavádí před vlastním vytvářením filtrovacích pravidel. Rozčlenění na moduly různého druhu má několik výhod. Jednak můžeme zavádět pouze ty, které opravdu potřebujeme, a samotné iptables jsou navrženy jako rozšiřitelné právě pomocí modulů jádra. Jako moduly lze přidávat jak cíle tak filtrovací pravidla.

Aby tyto moduly byly spravovatelné pomocí nástroje iptables, je i tento program navržen jako rozšiřitelný, a to pomocí dynamických knihoven. Každý netriviální modul, který vyžaduje předat nějaké parametry, má svůj doplněk v dynamické knihovně pro iptables. Vše je navrženo tak, aby implementace byla co nejjednodušší.

Rozčlenění na moduly má i další kladný dopad. Kód pro stavové filtrování, který byl původně součástí NAT (connection tracking) a implementován rovněž pomocí modulů, je nyní sdílen s iptables.

Při konstrukci filtrovacího pravidla můžeme zohlednit stav socketu, ke kterému procházející paket patří. Rozlišované stavy jsou tyto:

- NEW - jde o paket, který vytváří nový socket.
- ESTABLISHED - paket, který lze přiřadit některému již probíhajícímu spojení.
- RELATED - paket, který se vztahuje k nějakému probíhajícímu spojení, ale není jeho součástí (např. ICMP zpráva o chybě).
- INVALID - paket, který nelze přiřadit žádnému spojení. Tyto pakety by měly být likvidovány.

Budeme-li tedy např. chtít zvýšit odolnost systému proti stealth port scanům, můžeme to zajistit tím, že budeme likvidovat pakety s nastaveným FIN flagem a stavem INVALID.

Další možnost je zmíněna v předchozím odstavci. iptables umožňují rozlišovat jednotlivé flagy v TCP hlavičce a filtrovat pakety podle nich.

Dále je možné filtrovat podle TCP voleb nebo MAC adresy.

A konečně je zde pravidlo limit, které je možné použít například k omezení logování v některém řetězci. U toho pravidla můžeme nastavit maximální počet shod za sekundu a omezit tak přepřínování logu v řetězci, u kterého lze očekávat velký počet shod.

Existuje mnoho nástaveb, GUI a skriptů pro konfiguraci iptables přes přívětivější uživatelské rozhraní. Mezi nejznámější a nejrozšířenější patří například Shorewall.

1.1.1 Shorewall

Shorewall je sada skriptů a konfiguračních souborů, které definují pravidla přímo pro iptables. Aktivací daných pravidel se spustí a o vše se stará jádro Linuxu. Shorewall je určen pro jádra verze 2.4 a 2.6 a nedá se použít pro starší jádra, protože neobsahují netfilter/iptables.

2 DHCP SERVER

DHCP server (Dynamic Host Configuration Protocol) slouží k automatické konfiguraci TCP/IP na počítačích v síti. Klientská stanice vyše při startu operačního systému požadavek na konfiguraci, který je zachycen DHCP serverem. Ten pak vybere konfigurační parametry (IP adresu s příslušnou maskou podsítě a další volitelné parametry - např. adresu výchozí brány, adresy DNS serverů, jméno domény atd.) a přidělí je klientské stanici. Veškeré parametry pro klienty se nastavují pouze centrálně na serveru. Na jednotlivých stanicích stačí nastavit volbu, aby byly parametry TCP/IP konfigurovány automaticky z DHCP serveru.

DHCP server přiděluje klientům IP adresy z nadefinovaného rozsahu, a to většinou na určitou dobu (tzv. dobu pronájmu, angl. lease time). Před uplynutím doby pronájmu musí klient požádat o prodloužení, jinak bude po této době IP adresa považována za volnou a v případě nedostatku volných adres ji DHCP server přidělí jinému klientovi. Vše probíhá automaticky a pro uživatele zcela transparentně.

V DHCP serveru mohou být rovněž definovány tzv. rezervace (určitým klientům budou vždy přidělovány dané IP adresy). Adresa může být rezervována pro hardwarovou (MAC) adresu nebo jméno počítače. Tito klienti pak mají pevné IP adresy, které jsou konfigurovány automaticky. [4]

Hlavní výhody DHCP protokolu:

- výrazně nižší náročnost administrace (vše stačí nastavit pouze na serveru, není třeba konfigurovat jednotlivé stanice)
- eliminace mnoha potenciálních chyb (např. přidělení téže IP adresy dvěma různým stanicím, chybné nastavení výchozí brány na některé stanici apod.).
- správce sítě může "přečíslovat" síť nebo změnit vlastnosti sítě s minimálním zásahem do práce uživatelů

Funkce:

Klient komunikuje na UDP portu 68, server naslouchá na UDP portu 67.

Po připojení do sítě klient vyše broadcastem DHCPDISCOVER paket, na který odpoví DHCP server paketem DHCPOFFER s nabídkou IP adresy. Klient si z (teroreticky několika) nabídek vybere jednu IP adresu a o tu požádá paketem DHCPREQUEST a server mu

ji potvrdí odpovědí DHCPACK. Jakmile klient obdrží DHCPACK, může hned IP adresu a ostatní přidělená nastavení používat.

Klient musí před uplynutím doby pronájmu z DHCPACK obnovit svou IP adresu. Pokud lhůta uplyne aniž by dostal nové potvrzení, klient musí IP adresu přestat používat. [10]

3 CACHING PROXY SERVER

Proxy server slouží k tomu, aby se klient nepřipojoval na server, na kterém se zdroj nachází, ale aby se místo toho připojoval na proxy server, který požadovanou informaci získá a pošle ji klientovi. Proxy server, který je zároveň i tzv. caching (česky kešující), si informace přes něj procházející může uložit (do tzv. cache paměti – obvykle na HDD), a jsou-li po něm požadovány znovu, poskytne svou lokální kopii. Tím se výrazně urychlí komunikace, jelikož nakešovaný obsah se stahuje po lokální síti, která má obvykle vyšší rychlost přenosu než internetové připojení.

Proxy server samotný většinou slouží jako relativně bezpečné spojení dvou sítí, které mezi nimi umožňuje jen přesně definovanou komunikaci. Můžete například povolit HTTP/FTP přístup jen vybrané skupině lidí nebo počítačů k vybraným zdrojům. Počítače nemusí být schopné samy komunikovat s druhou sítí. Přínosem proxy-serveru je především možnost oddělení sítí a umožnění jen vybrané komunikace.

S kešujícím proxy serverem navíc můžete výrazně ušetřit provoz, který mezi sítěmi vzniká při opakovaném požadavku na stejný dokument. Velkým problémem kešování WWW je možná ztráta aktuálnosti poskytovaných dat - v keši je uložena starší verze dokumentu než na původním serveru. Při konfiguraci musíme najít kompromis mezi aktuálností dat a úsporou datového provozu.

3.1 Squid

Squid je vysoce výkonný kešující proxy server pro webové klienty. Podporuje FTP, gopher, a datové objekty HTTP. Na rozdíl od tradičních kešujících softwarů, Squid vykonává všechny požadavky v jednom procesu řízeném metodou I/O.

Squid si ponechává meta data, vysoce žádané objekty, uložené v paměti RAM. Implementuje také tzv. „negativní kešování“ neúspěšných požadavků.

Skládá se z:

- hlavního serverového programu **squid**,
- DNS (Domain Name System) vyhledávacího programu **dnsserver**,
- dalších volitelných programů pro přepisování požadavků a vykonávání autentizace,
- a spravovacích a klientských programů.

Jakmile Squid nastartuje, spustí nakonfigurovaný počet procesů *dns serveru*, z nichž každý může provádět tzv. Domain Name System (DNS) vyhledávání. Tím se redukuje čas potřebný na DNS vyhledávání, který cache potřebuje.

Typicky Squid použijeme k připojení vlastní sítě k Internetu. Squid je velmi výkonný, může pracovat s obrovskou zátěží a je často používán pro optimalizaci provozu na velkých páteřních sítích. Navíc se s ním dají dělat různé doplňkové operace, jako třeba odfiltrování reklamních bannerů...

4 AUTENTIZAČNÍ SERVER RADIUS

RADIUS (Remote Authentication Dial In User Service) je ochranný protokol na bázi „klient/server“, široce využíván k vzdálenému přístupu a autentizaci v podnikových sítích.

Je to server pro ověřování identity, zjištění oprávnění a evidenci (Authentication, Authorization, Accounting). Je primárně určen pro ISP (Internet Service Provider), ale může být použit pro jakékoliv internetové služby, které potřebují centrální zajištění správy uživatele.

Síť se dá nakonfigurovat tak, aby používala libovolnou autentizační metodu. Má podporu pro interní databázi, nebo může externě přistupovat pomocí modulů k MySQL, LDAP, Kerberos a spoustě dalších systémů. Radius běžně poskytuje podporu pro různé protokoly, jako je EAP, PAP, CHAP. Umí také spolupracovat s programovacími jazyky Perl a Python.

Informace o uživateli (uživatelské jméno a heslo a další), který se chce přihlásit do systému přijdou nejprve na NAS (Network Access Server), což je přístupový bod pro uživatele a stará se o komunikaci s uživatelem. NAS tyto informace přepośle Radiusu, který rozhodne zda uživatel má či nemá přístup. Z hlediska uživatele je veškerý autentizační proces transparentní.

Jak RADIUS server pracuje:

- Vyhledá údaje o uživateli
- Porovná údaje odeslané uživatelem s údaji v databázi
- Na základě správnosti těchto údajů, umožní uživateli přístup.

4.1 FreeRadius

FreeRADIUS je nejrozšířenějším open source RADIUS serverem. Je používán jak pro malé systémy, tak i pro systémy s milióny uživatelů. Je rychlý, flexibilní, konfigurovatelný a podporuje mnoho autentizačních protokolů (až 50).

5 VIRTUÁLNÍ PRIVÁTNÍ SÍŤ (VPN)

VPN umožňuje vytvořit přímé spojení mezi dvěma body, které může být zabezpečené šifrováním veškeré komunikace šifrovacími algoritmy (CHAP, MS-CHAP, MS-CHAPv2...). Dá se říci, že se na veřejné síti vytvoří mezi těmito dvěma body tunel. VPN se dá využívat k různým účelům - například k propojení dvou vzdálených sítí (dvě pobočky jedné společnosti v různých městech) nebo k připojení jednoho počítače do vzdálené sítě (zaměstnanec pracující doma se připojí do počítačové sítě společnosti, pro kterou pracuje). V prvním případě se dvě propojené sítě po vytvoření VPN chovají jako jeden celek. Ve druhém případě se počítač zaměstnance po vytvoření VPN stane součástí firemní sítě, tj. vypadá to, jakoby byl jeho vlastní počítač umístěn uvnitř počítačové sítě společnosti. [6]

Princip fungování virtuálních sítí spočívá v zapouzdření síťového paketu do dalšího paketu (původní data mohou být přitom také komprimována nebo šifrována). Takto upravený paket je poslán přes otevřenou síť adresátovi, u kterého je zase rozbalen. Virtuální privátní síť je možné realizovat různými způsoby - pomocí různých protokolů. Mezi nejznámější a nejpoužívanější z nich patří IP-tunelling, IPSEC a PPTP.

5.1 PPTPd (PoPToP)

Protokol PPTP je podporován ve většině operačních systémů (MS Windows 95/98/NT/2000/XP, Linux, aj.). Operační systémy MS Windows, které jsou mezi uživateli nejrozšířenější, obsahují implicitně podporu tohoto protokolu, takže odpadají problémy se sháněním instalací a složitou konfigurací. Na specifikaci PPTP protokolu se podílí společnost Microsoft. Hlavní silou PPTP je schopnost podporovat i jiné protokoly než IP. Hlavním nedostatkem je pak neexistence jednoho standardu pro šifrování a autentizaci. Proto dva produkty, které dodržují specifikace PPTP, mohou být vzájemně naprosto nekompatibilní, například pokud šifrují data různým způsobem.

Téměř vše z funkcionality protokolu PPTP je obstaráváno výhradně pomocí PPP (Point to Point Protocol), který momentálně podporuje autentizační metody PAP, CHAP, MSCHAP a MSCHAPv2 a několik šifrovacích a kompresních protokolů. Mezi nejpoužívanější šifrovací a komprimovací protokoly patří MPPE (Microsoft Point to Point Encryption) a MPPC (Microsoft Point to Point Compression). [6]

6 QUALITY OF SERVICE (QOS)

V poslední době dochází k rozvoji služeb, jejichž úspěšnost z pohledu uživatele významně závisí na časových charakteristikách komunikace přes počítačovou síť. Jde například o služby Internet telefonie, videokonference a další interaktivní služby. Uživatel požaduje, aby mu byla poskytnuta určitá definovaná kvalita služby (QoS). Například videosignál určitého rozlišení o určitém počtu obrázků za sekundu, zvuk určité šířky pásma nebo přenos souboru dat v určitém čase.

Zmíněné požadavky na QoS aplikací lze splnit, pokud se vhodným způsobem mapují na QoS počítačové sítě. Nejvýznamnější parametry, které definují QoS počítačové sítě jsou následující:

- Ztrátovost paketů - kolik procent paketů nedorazí od odesílatele k adresátovi,
- Průchodnost - objem dat v bajtech přenesený za jednotku času,
- Zpoždění - doba potřebná k přenosu paketu od odesílatele k adresátovi,
- Změna zpoždění - jak se mění zpoždění jednotlivých paketů během přenosu.

QoS počítačové sítě může být implementováno v různých vrstvách v rámci modelu počítačové sítě. Nejčastěji se používá implementace buď na úrovni ATM (je-li technologie ATM použita) nebo na úrovni protokolu IP (síťová vrstva). Při implementaci na úrovni protokolu IP existují dva hlavní přístupy řešení: integrované služby (integrated services, ve zkratce intserv) a rozlišované služby (differentiated services, ve zkratce diffserv). Historicky byla pozornost nejprve věnována QoS na úrovni ATM, později se pozornost přesunula na QoS na úrovni IP. Souvisí to s celkovým ústupem od používání technologie ATM. [2]

Problémy

V počátcích internetu nebyla velká nutnost mít QoS aplikace, takže v podstatě celý běžel na systému „maximálního vytížení“. V každé zprávě byly obsaženy 4 bity (informace o typu služby) a 3 bity (informace o prioritě), ale ty nebyly většinou vůbec využívány. Při cestě paketu ze zdroje k cíli mohou nastat nejrůznější komplikace a ty mají za následek následující problémy:

- **zahozené pakety** – routery nemusí doručit (drop) některé pakety, pokud dorazí v okamžiku, kdy jsou jejich zásobníky (buffery) plné. Některé, žádné, nebo všechny

pakety mohou být zahozeny (dropped). Toto závisí na stavu sítě a je nemožné předem rozhodnout co se stalo. Aplikace, která paket přijímá, si musí vyžádat přeposlání informace. To může mít za následek velké zpoždění v celkovém přenosu.

- **zpoždění** – než se paket dostane k cíli může trvat dlouho, protože se zasekne v dlouhých frontách, nebo si vybere okliku (nepřímou, delší cestu) aby se vyhnul zácpě. Alternativně může zvolit přímou, rychlou cestu. Zpoždění je velmi nepředvídatelné
- **„rozházené“ doručení** - když je soubor příbuzných paketů směřován skrz internet, každý paket může vybírat různé cesty, což má za následek různá zpoždění. Výsledek je ten, že pakety dorazí v jiném pořadí, než ve kterém byly odeslány. Tento problém vyžaduje speciální přídavné protokoly zodpovědné za uspořádání „rozházených“ paketů po té, co dorazí k cíli.
- **chyba** – někdy se stane, že pakety jsou doručeny jinam, zkombinovány dohromady, nebo poškozeny cestou. Příjemce toto musí zjistit a, jako kdyby byl paket zahozen (dropped), požádat odesílatele aby odeslání zopakoval. [11]

Traffic Shaping

Traffic shaping (tvarování provozu) zajišťuje mechanismus ke kontrole množství a objemu odeslaných dat do sítě (zaškrcení šířky pásma), a rychlosti při níž jsou data odesílána (limitování rychlosti). Z tohoto důvodu je nutno schémata traffic shapingu implementovat na rozhraní sítě a tím zajistit možnost, kontrolovat provoz vstupující do sítě. Může být nezbytné rozdělit dopravní tok v tzv. „bodě průniku“ (bod, ve kterém doprava vstupuje do sítě) do individuálních toků. To umožňuje kontrolnímu mechanismu traffic shapingu tvarovat data samostatně a s různými pravidly.

Existují dvě hlavní metody traffic shapingu: implementace leaky-bucket (děravý sektor) a implementace token-bucket (značení sektorů). Obě tyto schémata mají výrazně odlišné vlastnosti a každá je používána pro úplně jiné účely.

V počítačových sítích pracuje traffic shaping na základě zabránění „protržení“ dopravního toku, tzn. vyhlazení vrcholů a propadů při přenosu dat.

Názorný příklad práce traffic shapingu:

Před traffic shapingem: 10 paketů v první sekundě, 0 paketů v další sekundě, 10 paketů v další a 0 paketů v poslední sekundě.

Po traffic shapingu: 1 paket za 0.2 sekundy.

Výhody:

Výhody traffic shapingu při průchodu velkého množství dopravního toku přes „zúženou cestu“ (logickou nebo fyzickou) jsou:

- Menší „roztřepení“.
- Méně zahozených (drop) paketů.
- Menší prodleva.

Také je možno dosáhnout menších pingů, pokud je traffic shaping zkombinován s některým z druhů systému klasifikace pomocí Quality of Service.

6.1 HTB

HTB (Hierarchical Token Bucket) je srozumitelnější, intuitivnější a rychlejší náhradou za CBQ qdisc v Linuxu. Jak CBQ, tak i HTB pomáhají kontrolovat využití odchozí šířky pásma na daném spojení. Obojí umožňuje použít jedno fyzické spojení, které simuluje několik pomalejších spojení. V obou případech musíme specifikovat, jak rozdělit fyzické spojení na několik simulovaných a jak rozhodnout, které ze simulovaných spojení použít k odeslání daného paketu.

Stručný úvod do HTB systému:

Koncepčně je HTB náhodné číslo označených sektorů uspořádaných do hierarchie. Představme si nejjednodušší scénář. Základní pravidlo pro výpust fronty na libovolném zařízení se nazývá „root qdisc“.

Root qdisc bude obsahovat jednu třídu (komplexní řešení mohou obsahovat více tříd připojených k root qdiscu). Tato jedna HTB třída bude nastavena se dvěma parametry: rychlost (rate) a horní mez (ceil). Tyto hodnoty by měly být stejné jako u nejvyšší (nadřazené) třídy a reprezentují veškerou dostupnou šířku pásma připojení.

- **Rychlost (rate)** - v HTB znamená garantovanou šířku pásma pro danou třídu.
- **Horní mez (ceil)** – označuje maximální šířku pásma, kterou daná třída může využívat.

Jakákoliv šířka pásma používaná mezi frekvencí a horní mezí je půjčována z rodičovské třídy. Proto ta poznámka o velikosti frekvence a horní meze stejné jako u nejvyšší třídy.

Množství podřazených tříd může být pod touto třídou a každá z nich může mít přiřazenou nějakou šířku pásma, která je k dispozici od rodičovské třídy. V těchto podřazených třídách je nutné, aby hodnoty parametrů „rychlosti“ a „horní meze“ byly stejné jako hodnoty parametrů rodičovské třídy. Toto umožňuje rezervování určité šířky pásma určité třídě a také umožňuje HTB vypočítat poměr distribuce volné šířky pásma k poměru vlastních tříd.

HTB tedy implementuje mechanismus řazení do fronty podle tříd a zajišťuje uživateli kontrolu rozdělení celkové šířky pásma do tříd pomocí parametrů rychlosti a horní meze, jakožto i určovat poměr distribuce šířky pásma, pokud se uvolní extra šířka pásma (do výše horní meze).

V praxi by se měla šířka pásma nejvyšší třídy volit taková, jaká je celková šířka pásma k dispozici, minus její zlomková část. [1]

II.PRAKTICKÁ ČÁST

7 VÝBĚR VHODNÉHO SYSTÉMU

Při výběru vhodného operačního systému pro server tak, aby vyhovoval co nejvíce požadavkům zmíněným v zadání, jsem zkusil řadu možností. V následujících kapitolách popíšu výhody a nevýhody jednotlivých řešení, na které jsem v průběhu testování narazil.

Na začátku jsem zkoušel nakonfigurovat celý systém tak, aby běžel na tzv. WRAP desce, určené speciálně pro účely WiFi routeru. Z toho vyplynul počáteční výběr operačních systémů.

WRAP

WRAP je zkratka pro Wireless Router Application Platform.

WRAP deska je mini PC vhodné pro síťové aplikace jako je např.: router, firewall, VPN server atd. Jeho výhodou je, že neobsahuje zbytečné součásti jako např. zvukovou kartu, grafickou kartu, řadiče FDD apod., proto je, díky spolehlivosti, vhodná pro serverové nasazení.

Testoval jsem desku WRAP.2C, která má následující specifikace:

- **Procesor:** 266 MHz National Geode SC1100 CPU
- **Paměť RAM:** 128 MB
- **Úložiště dat:** veškeré aplikace a OS jsou uloženy na CompactFlash (CF) disku
- **Napájení:** 7~18V DC, příkon cca.3~5W při 12V DC včetně miniPCI karty, možnost napájení po ethernetu (PoE).
- **Vybavení:** Watchdog umístěn na CPU, LM77 teplotní čidlo pro monitorování CPU
- **Rozhraní:** 1x ethernet 10/100M (National DP83816), 2x miniPCI slot , 1x serial port (console). [9]

Ve dvou miniPCI slotech byly Atheros karty schopné vysílat signál na frekvenci 5GHz.

Pro WRAP desky jsou již vytvořeny speciální operační systémy na bázi Linuxu nebo BSD. Testoval jsem StarOS, m0n0wall a Debian-CZF.

7.1 m0n0wall

Je to projekt zaměřený na vytvoření kompletního softwarového balíku pro tzv. Embedded (zabudované) firewally. Při použití s embedded PC zajišťuje všechny důležité vlastnosti komerčních firewallů za zlomek jejich ceny.

M0n0wall je založený na OS FreeBSD a obsahuje balíky pro web server, PHP a další utility. Veškerá konfigurace systému je uložena v jediném XML souboru, což zajišťuje průhlednost a usnadňuje případné zásahy do systému. [5]

Má vytvořené grafické rozhraní v PHP, což na jednu stranu činí tento systém jednoduchý na nakonfigurování, ovšem na druhou stranu zásahy do funkčnosti systému jsou díky tomu řešení značně omezeny.

Stručný popis:

Po nainstalování OS na kartu CompactFlash a nastartování systému je možno přes sériovou konzoli nastavit IP adresu síťové karty a změnit heslo administrátora. Poté je již možno připojit se na m0n0wall server k administraci přes libovolný webový prohlížeč zadáním IP adresy serveru. Zobrazí se GUI rozhraní (Obr. 2), kde je možno nastavit veškeré potřebné parametry, jako například IP adresy síťových karet, firewallu a jeho pravidel, jednoduchého traffic shapingu, DNS a DHCP serveru a také VPN serveru, který zvládá jak protokol PoPToP, tak IPSec.

Výhody:

- jednoduchost (nastartování přes grafické rozhraní)
- mnoho implementovaných funkcí
- cena (je zcela zdarma)

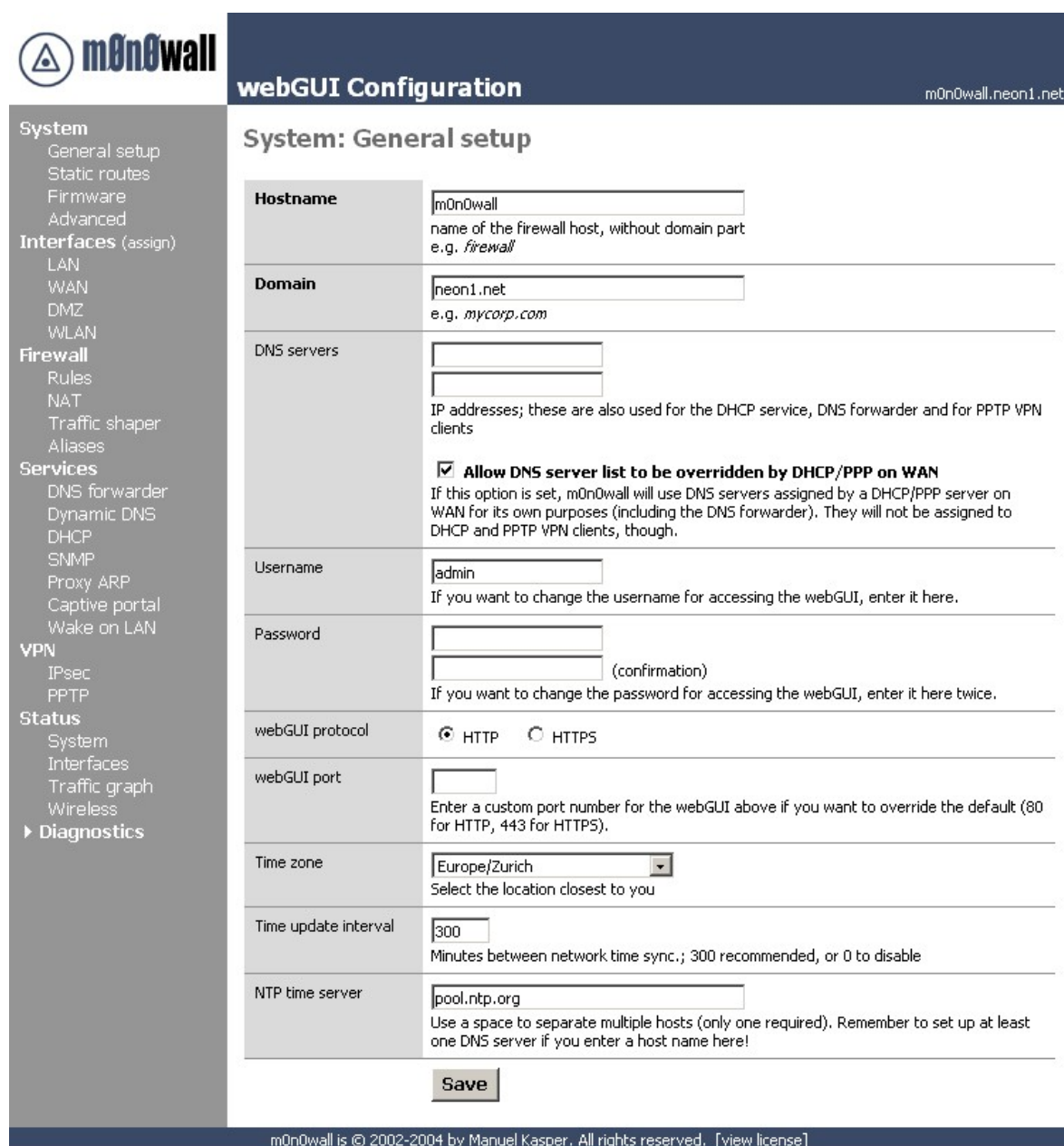
Nevýhody:

- omezená možnost konfigurace (pracnost nastavení „nestandardních“ funkcí mimo GUI)
- použití horšího a zastaralejšího QoS traffic shapingu na bázi CBQ namísto novějšího a propracovanějšího HTB

Závěr:

Z hlediska zadání mé práce a s přihlédnutím k případnému nesnadnému přidání neimplementovaných funkcí, jsem možnost použití tohoto jinak perfektního systému (z hlediska

jak funkcionality, tak jednoduchosti) navrhl. Použití metody HTB je totiž pro mne nutnost a kompilace nového jádra systému FreeBSD by byla neúměrně pracná.



The screenshot shows the m0n0wall webGUI Configuration page. The left sidebar contains a navigation menu with categories: System (General setup, Static routes, Firmware, Advanced), Interfaces (assign) (LAN, WAN, DMZ, WLAN), Firewall (Rules, NAT, Traffic shaper, Aliases), Services (DNS forwarder, Dynamic DNS, DHCP, SNMP, Proxy ARP, Captive portal, Wake on LAN), VPN (IPsec, PPTP), and Status (System, Interfaces, Traffic graph, Wireless, Diagnostics). The main content area is titled 'webGUI Configuration' and 'System: General setup'. It contains several configuration fields: Hostname (m0n0wall), Domain (neon1.net), DNS servers (empty), Username (admin), Password (empty), webGUI protocol (radio buttons for HTTP and HTTPS), webGUI port (empty), Time zone (Europe/Zurich), Time update interval (300), and NTP time server (pool.ntp.org). A 'Save' button is at the bottom. The footer indicates 'm0n0wall is © 2002-2004 by Manuel Kasper. All rights reserved. [view license]'.

Obr. 1. Vzhled grafického rozhraní m0n0wallu.

7.2 StarOS

StarOS je komerční operační systém určený mimo jiné pro WRAP desky. Je založen na OS Linux a jeho konfigurace je řešena přes grafické rozhraní. Může být použit pro různé aplikace:

- Chová se jako klientská jednotka, což umožňuje koncovému uživateli přistupovat k internetovému připojení z více než jednoho PC buď se standardní, nebo NAT IP adresou.
- Chová se jako AP (Access Point) jednotka, což umožňuje bezdrátové připojení a směrování na straně ISP (Internet Service Provider – Poskytovatel Internetového Připojení)
- Chová se jako IS (Internet Server) s následujícími službami: DNS, Transparentní caching Proxy, Routing, NAT, DHCP, Firewall a traffic shaping.
- Chová se jako router (směrovač) s následujícími službami: Routing, NAT, DHCP, Firewall a traffic shaping

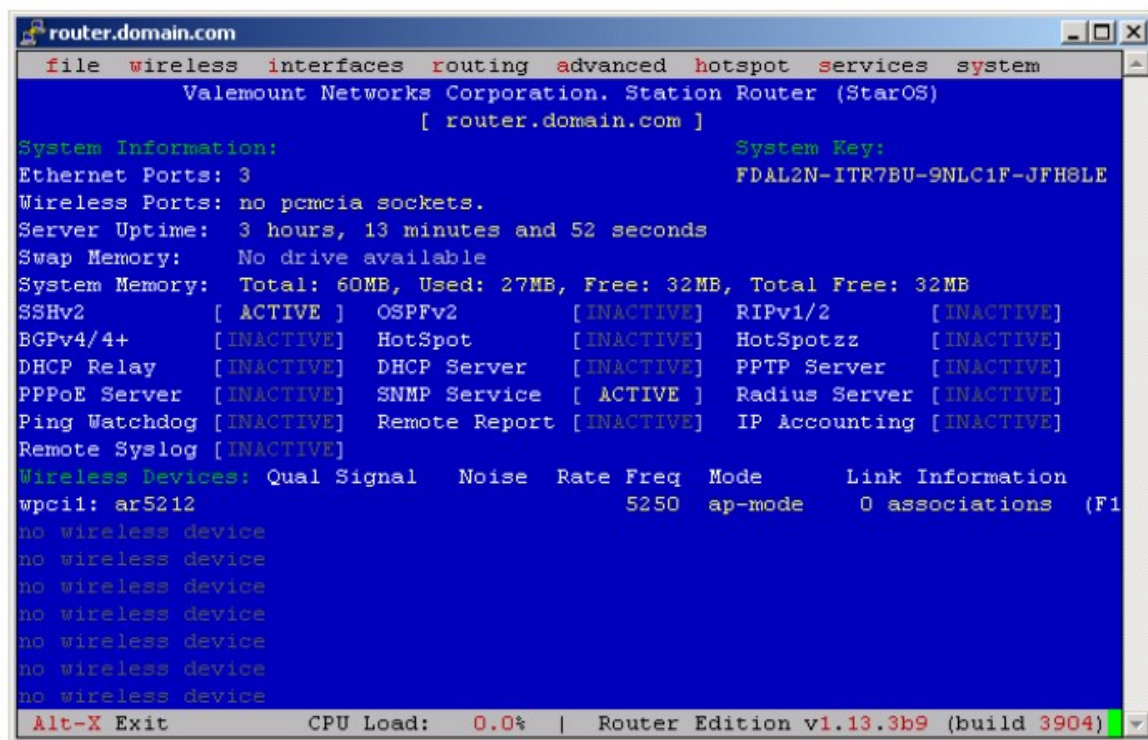
Vlastnosti (zajímavé z hlediska zadání mé práce):

- Traffic shaping založen na IP adrese a/nebo protokolu
- Firewall může být nakonfigurovaný přes skriptovací rozhraní
- Možnost nastavení velkého množství statických i dynamických NAT pravidel
- Měřič síly signálu pro snadné nastavení antény
- Schopnost vytváření různých statistik
- Vzdálený upgrade firmwaru
- DHCP server
- DHCP Proxy – předává DHCP požadavky vzdálenému DHCP serveru
- SSH (v2) server kvůli možnosti vzdálené konfigurace
- Proxy Server s možností cachování Web a FTP.
- DNS server
- “Captive Portal” webové přihlašování
- PPPoE, PPTP Tunelovací protokoly
- Radius server a autentizace pro PPPoE, PPTP [7]

Stručný popis:

Po nainstalování systému a následném spuštění a nastavení IP adresy síťové karty a přihlašovacího hesla administrátora, je možno se přihlásit na server prostřednictvím SSH pro-

tokolu. Po úspěšném přihlášení do systému se objeví konzolové konfigurační rozhraní. (Obr. 2). Po přihlášení je nutno zadat zakoupený registrační klíč, jinak je server plně k dispozici jen 24 hodin. Po té se restartuje a všechny konfigurační změny smaže. Konfiguraci není možné ani zálohovat.



Obr. 2. Grafické rozhraní StarOSu

Výhody:

- velké množství implementovaných funkcí
- relativně snadná konfigurace v kombinaci s možností konfigurace náročnějších a komplexnějších serverových řešení

Nevýhody:

- komerční software. (Cena okolo 900 Kč)
- uzavřenost systému (nemožnost upravit a vložit neimplementované funkce)

Závěr:

Z mnoha hledisek je to skvělý systém, který byl horkým kandidátem na zavedení do provozu. Hlavní příčinou toho, proč se tak nestalo byla cena systému a nedostatek času pro intenzivnější testování, které by bylo důležité pro to, aby se nekupoval tzv. „zajíc v pytli“.

7.3 Debian-CZF

Je to tzv. „minidistribuce“ Debian GNU/Linux pro CZFree.Net.

Vlastnosti:

- Bootovatelné CD, konfigurace se dá měnit připálením další session na CD.
- Velikost cca 120 MB, vhodné pro flash disky, USB klíčenky, atd.
- Read-only filesystem, zápis pouze na swapovatelný ram-disk.
- Balíčky z Debian Sarge.
- Firewall, Shaper, Bridge, Quagga (OSPF), DHCP, DNS, HostAP, MADWiFi, mc, vi, atd.
- Vzorové konfigurační soubory sítě, Quaggy, DHCP, DNS, firewallu, atd. [3]

Výhody:

- otevřenost systému (přece jenom je to plnohodnotná distribuce Debian Linuxu upravená pro účely WiFi serveru)
- přístup k veškerým konfiguračním souborům s možností nahrát si vlastní balíky a kompilace jádra

Nevýhody:

- špatná podpora desek WRAP => nutnost nastavit, „opatchovat“ a zkompilovat jádro

Závěr:

Z hlediska zadání mé bakalářské práce je tento systém nevyhovující, protože je již přednastaven na síť v rámci CZFree.net a tudíž by bylo zhruba stejně pracné přenastavovat veškeré parametry ručně na jinou síť, jako nastavit celý systém od začátku.

Navíc po konečném rozhodnutí využití místo desky WRAP běžného PC jako serveru, jsem zvolil instalaci plnohodnotného Debian GNU/Linux s tím, že celý systém nakonfiguruji sám od začátku.

8 KONFIGURACE SERVERU (DEBIAN GNU/LINUX)

Operačním systémem je Debian GNU/Linux (verze 3.1 Sarge).

Tento operační systém je založen na jádře Linux. Je to jedna z nejrozšířenějších linuxových distribucí a je známa vysokou stabilitou a bezpečností a hodně se proto instaluje na servery. Obsahuje mnoho balíků a serverových aplikací. Všechny programy až na Prometheus QoS použité v mé práci jsou součástí Debianu.

Dále budu pokračovat popisem jednotlivých programů, které byly nutné pro funkčnost systému tak, aby vyhovovala zadání. Důležité části konfiguračních souborů budou uvedeny také, spolu s případnými poznámkami k jednotlivým parametrům.

8.1 Firewall & Masquerade

Jako firewall jsem zvolil Shorewall, jehož funkci jsem popsal v teoretické části.

Zde bych rád popsal koncept firewallu na tomto serveru. Celá lokální síť je „schovaná“ za jednou veřejnou IP adresou a metodou tzv. Masquerade je vyřešen překlad adres (NAT). V serveru jsou dvě síťové karty. První (eth0) je ta, kterou je přiveden internet od providera. Druhá (eth1) slouží k „rozvodu“ internetu do lokální sítě, ke které jsou připojeny počítače uživatelů.

Moje konfigurace zahrnuje blokování veškerého příchozího spojení z internetu do vnitřní sítě a zároveň povoluje přístup na některé porty na serveru z internetu (Tab. 1). Povolené porty pro připojení z lokální sítě na server, jsou uvedeny v tabulce č.2.

Tab. 1. Seznam povolených portů z internetu na server:

Číslo portu	Služba	Relevantní část z konfiguračního souboru (/etc/shorewall/rules)
8	Ping	ACCEPT net fw icmp 8
20	FTP-data	ACCEPT net fw tcp 20
21	FTP	ACCEPT net fw tcp 21
22	SSH	ACCEPT net fw tcp 22
25	SMTP – mail	ACCEPT net fw tcp 25
80	Apache Web Server	ACCEPT net fw tcp 80
110	POP3	ACCEPT net fw tcp 110

Tab. 2. Seznam povolených portů z lokální sítě na server:

Číslo portu	Služba	Relevantní část z konfiguračního souboru (/etc/shorewall/rules)
8	Ping	ACCEPT loc fw icmp 8
21	FTP	ACCEPT loc fw tcp 21
22	SSH	ACCEPT loc fw tcp 22
25	SMTP – mail	ACCEPT loc fw tcp 25
53	DNS server	ACCEPT loc fw udp 53
67	DHCP server	ACCEPT loc fw udp 67
80	Apache Web Server	ACCEPT loc fw tcp 80
110	POP3	ACCEPT loc fw tcp 110
1723	VPN server	ACCEPT loc fw tcp 1723
1812	Radius Authenticat.	ACCEPT loc fw tcp 1812
1813	Radius Accounting	ACCEPT loc fw tcp 1813
1814	Radius Proxy	ACCEPT loc fw tcp 1814

Masquerade – v praxi to znamená, že jakýkoliv počítač na vnitřní síti, který přistupuje do internetu bude navenek vystupovat pod tzv. „vnější“ IP adresou (adresou serveru). Tím jsou ochráněny všechny počítače ve vnitřní síti, protože případný útočník nemá přístup přímo k jednotlivým jejím počítačům.

Relevantní část konfiguračního souboru (/etc/shorewall/masq):

```
#INTERFACE SUBNET ADDRESS
eth0 eth1
```

Výňatek dalších relevantních částí konfiguračních souborů z /etc/shorewall/:

- zařízení (/etc/shorewall/interfaces):

```
#ZONE INTERFACE BROADCAST OPTIONS
net eth0 detect routefilter,dhcp,tcpflags
loc eth1 detect tcpflags
```

➤ eth0 je nastavena tak, že IP adresu jí přiděluje DHCP server, eth1 má nastavenou statickou IP.

- zóny (/etc/shorewall/zones):

```
#ZONE DISPLAY COMMENTS
net Net Internet
loc Local Local Network
```

➤ net je vstupní brána internetu (přiřazená na eth0) a loc je lokální síť (přiřazená na eth1). VPN je kvůli umožnění připojení na VPN server.

- politika (/etc/shorewall/policy):

#SOURCE	DEST	POLICY	LOG	LEVEL	LIMIT:BURST
loc	net	ACCEPT			
fw	net	ACCEPT			
net	all	DROP	ULOG		
all	all	REJECT	ULOG		

- Definuje základní politiku firewallu. Nejprve je zakázána veškerá komunikace směrem z internetu na veškeré zóny a naopak veškeré pakety z lokální sítě na internet a z firewallu na internet jsou povoleny.

8.2 Squid

Jako caching proxy server jsem zvolil Squid. Nastavil jsem jej tak, že funguje jako tzv. transparentní proxy server. To znamená, že ze strany klienta není nutné žádné speciální nastavení a klient ani neví, že veškerá jeho komunikace se serverem přes port 80 (surfování na internetu) běží přes caching proxy server Squid. Tedy „transparentně“.

Aby bylo tohle umožněno, je nutno do konfiguračního souboru Shorewallu zadat přesměrování všech požadavků připojení na www (od lokálních uživatelů na internet) na port 3128 (na kterém poslouchá Squid). Tím je zajištěno, že veškeré surfování uživatelů bude cachováno a nelze jej nijak obejít.

Relevantní část konfiguračního souboru Shorewallu (/etc/shorewall/rules):

REDIRECT	loc	3128	tcp	www	-	!192.168.61.73
ACCEPT	fw	net	tcp	www		

- Touto konfigurací je ošetřen i případné nechtěné cachování při přístupu na internet přímo z firewallu. (!IP_serveru).

Zajímavé parametry z konfiguračního souboru Squidu (/etc/squid/squid.conf):

```
cache_dir ufs /var/spool/squid 100 16 256
```

- Udává mimo jiné umístění a velikost cache paměti na disku (/var/spool/squid a velikost 100MB).

```
http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
acl our_networks src 192.168.1.0/24 192.168.2.0/24
http_access allow our_networks
http_access allow localhost
http_access deny all
http_reply_access allow all
```

- Tyto parametry v podstatě definují přístupová práva. Umožňuje management z localhost a blokuje všechny ostatní, zakazuje všechny porty, které nejsou v seznamu Safe_ports, zakazuje připojování na SSL port, definuje rozsah IP adres lokální sítě a povoluje přístup pouze z nich.

8.3 DHCP server

DHCP server je prozatím nastaven tak, aby přiděloval klientským stanicím IP adresu z rozsahu 192.168.1.10 až 192.168.1.20 a masku podsítě 255.255.255.0. Dále přiřazuje výchozí bránu a DNS server na IP adresu příslušného serveru. Nastaví také výchozí dobu pronájmu na 600 sekund a maximální dobu pronájmu na 7200 sekund.

Konfigurační soubor (/etc/dhcpd.conf) vypadá následovně:

```
#option domain-name "";
option domain-name-servers 192.168.1.2;
option routers 192.168.1.2;

option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;

# lokalka
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.20;
}
```

8.4 VPN

Pro VPN server byla jako jeden z požadavků stanovena jednoduchost a minimalizace nastavení na straně klienta. Předpokládal jsem, že klienti přistupující k VPN serveru budou ve většině případů používat operační systém Microsoft Windows.

VPN je realizováno prostřednictvím protokolu PPTP a jako pptp server byl vybrán projekt PoPToP, který využívá ppp protokol. Pro autentizaci byl zvolen RADIUS, konkrétně Free-RADIUS, který komunikuje s SQL databází uživatelů, a to MySQL. Na serveru je také spuštěn HTTP server Apache, které budou uživatelům poskytovat veškeré informace týkající se nastavení, registrace, plateb a podobných záležitostí důležitým pro připojení do sítě.

V konfiguračním souboru Shorewallu bylo nutné udělat tyto změny:

- tunely (/etc/shorewall/tunnels):

```
pptpserver    net      0.0.0.0/0
pptpserver    loc      0.0.0.0/0
```

➤ tyto dva řádky umožňují vytvoření tunelu z internetu i z lokální sítě

- definice nové zóny (/etc/shorewall/zones):

```
#ZONE    DISPLAY    COMMENTS
vpn      VPN        Remote Users
```

➤ zde je definována nová zóna, do které budou přiřazeni uživatelé připojení na VPN server. Tímto je možné určit případná další pravidla a politiku týkající se klientů připojených k VPN.

- povolení GRE protokolu (/etc/shorewall/rules):

```
ACCEPT      loc      fw      47      -
```

➤ *povolení protokolu GRE (port 47) z lokální sítě na server.*

Dále kvůli podpoře *mppe* šifrování, bylo nutné přidat do jádra systému příslušný patch a celé jádro překompilovat.

8.4.1 PPTPD

Server pptpd je nastaven tak, že celé připojení je šifrováno protokolem MPPE (při autentizaci pomocí MSCHAPv2), což zaručuje 100% kompatibilitu se stanicemi postavenými na operačním systému MS Windows.

Uživatel, který se připojí na VPN server je automaticky autentizován přes RADIUS server.

Relevantní část konfiguračního souboru (/etc/ppp/pptpd-options):

```
name pptpd
ms-dns 192.168.1.2
#debug
#dump
local
lock
asyncmap 0
-chap
-mschap
+mschap-v2
require-mppe
lcp-echo-failure 30
lcp-echo-interval 5
ipcp-accept-local
ipcp-accept-remote
plugin radius.so
```

➤ *parametr „plugin radius.so“ zajišťuje nahrání modulu RADIUS, a přes něj je poté prováděna autentizace uživatele*

Dále konfigurační soubor /etc/pptpd.conf:

```
option /etc/ppp/pptpd-options

localip 192.168.1.2
remoteip 192.168.1.50-253
```

➤ *adresu do vnitřní VPN sítě uživateli přiřadí z rozsahu 192.168.1.50 až 192.168.1.253*

8.4.2 FreeRADIUS

Jak jsem zmínil výše, jako autentizační server byl vybrán FreeRADIUS. Ten je napojen na MySQL databázi. MySQL server, stejně jako FreeRADIUS a všechny ostatní služby, je nainstalován stejném PC.

Po připojení klienta na VPN, je pppd serverem spuštěn *radius.so plugin* (viz. výše), který spouští *radiusclient*. Ten posílá požadavek na server FreeRADIUS, který na základě požadavku vyhodnotí údaje přijaté od klienta a na základě správnosti těchto údajů je mu přidělen/zamítnut přístup.

Zajímavé části konfiguračního souboru:

```
modules {
    mschap {
        authtype = MS-CHAP
        use_mppe = yes
        require_strong = yes
    }
    authorize {
        preprocess
        sql
        mschap
    }
    authenticate {
        Auth-Type MS-CHAP {
            mschap
        }
    }
    preacct {
        preprocess
        files
    }
    accounting {
        acct_unique
        detail
        sql
    }
    session {
        sql
    }
}
```

- *tohle zajistí, že RADIUS použije MySQL server pro databázi uživatelů a zároveň jako způsob autentizace použije MSCHAP a MPPE šifrováním.*

8.5 HTB – Prometheus QoS

Je určen ke snadné manipulaci s traffic shapingem. Umožňuje návrh mnoha schémat pro spravedlivé rozdělování šířky pásma podle IP adresy a datových limitů.

Je to program napsaný v jazyce C a potřebuje ke správné funkci Linuxové jádro zkompilevané s podporou HTB a SFQ QoS. Proto jsem musel jádro ještě jednou zkompileovat a zahrnout do něj i tyto moduly.

Program využívá souboru /etc/hosts pro nastavení šířky pásma pro jednotlivé IP adresy.

Relevantní část tohoto souboru je zde jako příklad:

```
192.168.1.13      Jakub      #via-prometheus-32-64
```

➤ důležitá je část za „#“. První číslo udává rychlost (rate) a druhé horní mez (ceil)

Hlavní konfigurační soubor se nachází v /etc/prometheus.conf, kde se nastavují všechny základní parametry jako třeba názvy zařízení a jejich přiřazení (WAN, LAN), cesty k programům (tc, iptables), základní rychlosti přidělené ISP, cesty k logům atd.

ZÁVĚR

Měl jsem za úkol nakonfigurovat server linuxového typu tak, aby byl vhodný pro tzv. „routování“ internetu k připojeným klientům. Na základě požadavků na funkce serveru, jsem použil tyto programy: Shorewall, FreeRADIUS, MySQL, Apache, PPTPD, Squid, Prometheus QoS a PureFTPD.

Shorewall je tzv. „firewall“, starající se o zabránění nežádoucím uživatelům, hackerům, virům a červům v přístupu do sítě. FreeRADIUS je napojen na MySQL databázi a dohromady slouží k autentizaci uživatelů připojících se do sítě. Apache je web server a poskytuje uživatelům www stránky od ISP. Squid se stará o cachování obsahu www stránek a ve výsledku zrychluje surfování po internetu. VPN je postavena na protokolu PPP a serveru PPTPD a je zabezpečena šifrováním. Programem Prometheus QoS je zabezpečeno spravedlivé rozdělování šířky připojení mezi uživatele.

Z výčtu použitých programů je zřejmé, že funkce serveru je koncipována s ohledem na co největší uživatelské pohodlí a bezpečnost. Pohodlí jsem zajistil tím, že maximální počet služeb běží transparentně, tzn. že uživatel nemusí nic nastavovat. Při konfiguraci serveru pro WiFi připojení je nutné zohlednit zejména bezpečnost a proto jsem kladl největší důraz na šifrování přenosu mezi serverem a klienty. To je zajištěno MPPE šifrováním při připojení na VPN server a autentizací pomocí MS-CHAPv2. Jelikož přístup do internetu je možný až po připojení na VPN server, je proto veškerý přenos šifrovaný a tudíž velice těžce odposlechnutelný potenciálním útočníkem.

Jako doplňkovou službu jsem nakonfiguroval i FTP server (PureFTPD) napojený na MySQL databázi. FTP server zajišťuje jednoduché sdílení souborů a umožňuje přístup pouze registrovaným uživatelům.

SEZNAM POUŽITÉ LITERATURY

- [1] Martin Devera *HTB manual - user guide* [online]. 2005, [2005-06-06].
<http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm>
- [2] *Cesnet - QoS* [online]. 2005, [2005-06-06]. <http://www.cesnet.cz/qos/>
- [3] *CZFree.Net Brno* [online]. 2005, [cit. 2005-06-02]. <http://wiki.webstep.net/czfree/2>
- [4] *Kerio: DHCP server* [online]. 2005, [2005-06-02].
<http://www.kerio.cz/manual/kwf/cz/ch04s04.html>
- [5] *m0n0wall* [online]. 2005, [2005-06-01]. <http://www.m0n0.ch/wall/>
- [6] *Masarykova Univerzita - VPN Server* [online]. 2005, [2005-06-05]. <http://www.ics.muni.cz/bulletin/issues/vol14num02/pesa/pesa.html>
- [7] *StarOS - Dokumentace* [online]. 2005, [2005-06-06]. <http://www.staros.com>
- [8] *Stavíme Firewall* [online]. 2005, [2005-06-03]. <http://www.root.cz/clanky/stavime-firewall-1/>
- [9] *Sys-DataCom - wrap boardy* [online]. 2005, [2005-06-04]. http://www.sys-data.com/shop.php?action=podrobnosti&cislo_polozky=25&kategorie=Wrap%20boardy
- [10] *Wikipedia - DHCP* [online]. 2005[2005-06-05]. <http://cs.wikipedia.org/wiki/DHCP>
- [11] *Wikipedia - Quality of Service* [online]. 2005, [2005-06-06].
http://en.wikipedia.org/wiki/Quality_of_service

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AP	Access Point
BSD	Berkeley Software Distribution
CBQ	Class Based Queuing
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EAP	Extensible Authentication Protocol
FTP	File Transfer Protocol
GUI	Graphical User Interface
HDD	Hard Disk Drive
HTB	Hierarchical Token Buckets
HTTP	Hyper Text Transfer Protocol
CHAP	Challenge-Handshake Authentication Protocol
IMAP3	Internet Message Access Protocol
IP	Internet Protocol
IPSec	Secure Internet Protocol
IPX	Internetwork Packet Exchange
ISP	Internet Service Provider
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MPPC	Microsoft Point-to-Point Compression
MPPE	Microsoft Point-to-Point Encryption
NAS	Network Access Server
NAT	Network Address Translation/Translator
PAP	Password Authentication Protocol
POP3	Post Office Protocol
PPP	Point-to-Point Protocol
PPTP	Point to Point Tunneling Protocol
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Server/Service
SQL	Structured Query Language
SSH	Secure Shell
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VPN	Virtual Private Network
WRAP	Wireless Router
XML	eXtensible Markup Language

SEZNAM OBRÁZKŮ

Obr. 1. Vzhled grafického rozhraní m0n0wallu.....	24
Obr. 2. Grafické rozhraní StarOSu.....	26

SEZNAM TABULEK

Tab. 1. Seznam povolených portů z internetu na server:.....	28
Tab. 2. Seznam povolených portů z lokální sítě na server:.....	29