ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

FAKULTA PROVOZNĚ EKONOMICKÁ

INFORMATIKA A SYSTÉMOVÉ INŽENÝRSTVÍ

Katedra informačního inženýrství

Router s operačním systémem Linux

Router with Linux operating system

Bakalářská práce

Autor: Jakub Sojka

Vedoucí práce: doc. Ing. Arnošt Veselý, CSc.

Praha březen 2010

Čestné prohlášení:

Čestně prohlašuji, že jsem bakalářskou práci na téma „Router s operačním systémem Linux“ vypracoval samostatně za použití uvedené literatury a po odborných konzultacích s doc. Ing. Arnoštem Veselým, CSc.

V Praze dne 30.března 2010

…….

Jakub Sojka

Poděkování:

Chtěl bych poděkovat doc. Ing. Arnoštu Veselému, CSc., vedoucímu mé bakalářské práce, za poskytnutí potřebných informací pro vypracování.

Souhrn

Cílem této práce je představení operačního systému Linux v roli routeru, vyzdvihnout vlastnosti Linuxu v této roli a seznámit uživatele s konfigurací základních služeb.

První část se zabývá představením distribucí a prostor je věnován také instalačním procedurám.

Druhá část se zabývá základním nastavením sítě, DHCP a DNS.

Třetí část je o bezpečnosti, především o bezpečném vzdáleném přístupu a konfiguraci firewallu.

Vlastním přínosem je konfigurace čisté Linuxové distribuce pro popsané služby s využitím získaných teoretických poznatků.

Klíčová slova

linux, distribuce, instalace, DHCP, DNS, směrování, SSH, bezpečnost, firewall, iptables

Summary

The purpose of this work is to introduce Linux operating systém in a role as router, to highlight features of Linux in this role and to inform user about configuration of basic services.

First part is handling introduction to Linux distributions and some space is devoted to installation proceduers too.

Second part is handling basic configuration of network, DHCP and DNS.

Third part is about security, in the first place about secure remote acces and configuration of firewall.

The contribution of this work is configuration of clear Linux distribution for described services by using acquired theoretical knowledge.

Key words

linux, distribution, installation, DHCP, DNS, routing, SSH, security, firewall, iptables

Obsah

Úvod

Termín Linux se pro tento operační systém již natolik ujal, že se používá jako název. Ovšem je to alespoň z poloviny nepřesné, protože správné pojmenování je GNU/Linux. Linux je označení pro jádro operačního systému, zatímco GNU ukazuje na vlastní operační systém.

Linux jako jádro vznikal na začátku 90. let 20. století, zásluhou mladého finského studenta Linuse Torvaldse. Linusovi se tehdy velmi líbil unixový operační systém Minix, ke kterému ale nebylo možné získat zdrojové kódy. Velké unixové operační systémy byly zase nad studentovi finanční možnosti, proto se rozhodl psát kódy vlastní. Jeho jádro si okamžitě získalo patřičnou pozornost řady příznivců, a díky tomu se vývoj rozrostl do nevídaných rozměrů. Linus následně uvolňuje svůj systém pod licencí GNU GPL.

Projekt GNU (rekurzivní akronym GNU's Not Unix) inicioval na začátku osmdesátých let Richard Matthew Stallman. Hlavní myšlenkou projektu je, že každý software by měl být svobodný. Není tím myšlena nutnost platit za software, jako spíš dodávání softwaru společně s jeho zdrojovými kódy. Z toho pak vzniká pár dalších myšlenek: uživatelé nemusí jednat s vývojářem, který danému softwaru ani nemusí poskytovat podporu. Kód, který vyvíjí a kontroluje větší skupina, či dokonce několik skupin programátorů, je kvalitnější než ten, který si vyvíjí jedna společnost. A hlavně myšlenka samotných uživatelů, kteří pokud potřebují nějakou funkci, tak ji mohou do daného softwaru přidat a následně ji ještě poskytnout dále, aby ji mohli využívat i další uživatelé.

Pod GNU následně vzniká tzv. Veřejná licence GNU (GNU General Public Licence – GNU GPL). Zdrojové softwarové kódy, které vzniknou pod GPL, musí být nadále šířeny pod GPL licencí – GPL jasně říká, že programátor má respektovat svobodu druhých. Daný software může být i prodáván, avšak prodávající se zavazuje k poskytnutí zdrojových kódů, což GPL popisuje jako stejnou svobodu druhých, kterou měli i ti, jež daný kód upravili. Další neméně důležitou součástí jsou podmínky zodpovědnosti. Říkají, že programátor nenese odpovědnost za škody, které by mohly vzniknout v důsledku užívání jeho softwaru.

Během deseti let se lidem okolo projektu GNU podařilo vytvořit všechny potřebné nástroje (aplikace, systémové knihovny, textový editor apod.) a v roce 1990 začaly práce na jádru operačního systému, které by zajišťovalo komunikaci s hardwarem a samotnými nástroji. Jádro se jmenovalo Hurd. V roce 1991 přichází na scénu Linus se svým jádrem, jehož vývoj byl daleko rychlejší než vývoj Hurdu právě díky řadě příznivců, kteří na Linuxu spolupracovali. Následně dochází k logickému kroku a operační systém GNU se spojuje s jádrem Linux. Tím vzniká finální produkt GNU/Linux, což je také správné pojmenování.

Cíl práce a metodika

Cílem práce je náhled na efektivní využití Linuxu jako operačního systému pro router. Text se soustředí především na popis a konfiguraci stroje pod tímto operačním systémem; zhodnocení vlastností operačního systému Linux v roli routeru a v prostředí domácnosti či malé firmy a charakteristiku vybraných služeb a základní zabezpečení proti možným útokům.

V první části se práce soustředí na operační systém Linux v roli routeru. Především se pak zabývá vlastnostmi daného stroje - zhodnocení toho, co nám může Linux v takovéto roli poskytnout. Dále první kapitola představuje distribuce, jejich krátké zhodnocení a samotnou instalaci zvolené distribuce.

Druhá kapitola se soustředí na vybrané intranetové služby, jejich fungování z pohledu sítě a konfigurace pod vybranou linuxovou distribucí. Zmíněno bude základní nastavení sítě, dále dvě základní služby - DHCP (Dynamic Host Configuration Protocol) a DNS (Domain Name Systém). Nastavení těchto služeb a tím pádem příprava stroje na další využití.

Poslední část se soustředí na zabezpečení. Zabývá se obecnými zásadami bezpečnosti, vzdáleným přístupem v případě nutnosti vzdálené konfigurace a představením Linuxového firewallu.

Operační systém Linux

Operační systém lze využít různými způsoby. Linux, Windows, MacOS, Unix, či jiný univerzální operační systém lze použít jako pracovní stanici, router, poštovní server, firewall a mnoha dalšími způsoby. Práce se zaměřuje na využití Linuxu jako routovacího zařízení, které bude poskytovat i jiné služby pro malou síť.

Vlastnosti Linuxu jako routeru

K plánované úloze Linuxu patří určité požadavky. Ty by měl být daný systém schopen splnit. Mezi požadavky patří:

Spolehlivost

Linux má pověst velmi spolehlivého systému nejenom díky své stabilitě. Ale právě stabilita je u routovacího zařízení, které by mělo pracovat ve funkci edge-routeru, nutná. Plánované zařízení totiž poběží neustále a nestabilita by znamenala více nákladů na správu.

Software

Pro Linux existuje celá řada kvalitních a zástupných programů určených pro serverové prostředí. Celá řada z nich byla napsána přímo na Linux.

Cena a licence

Linux lze bezplatně stáhnout z internetu, pokud nepočítáme náklady spojené s připojením. Pořizovací cena Linuxu je tedy nulová, což je příjemné z hlediska celkových nákladů.

Licence pro Linux je GNU/GPL, jak již bylo zmíněno výše. Tato licence se ovšem vztahuje obvykle i na ostatní součásti linuxové distribuce (pokud ne, tak bývají vydávány pod jinými volnými licencemi - změny většinou doznává jen politika ohledně šíření upraveného kódu).

Bezpečnost

Linux jako operační systém je odolný proti různým virům, či wormům, hlavně díky tomu, že většina těchto programů je psána na konkurenční systém Windows. Síťové bezpečnosti, která je složitější, je věnována samostatná kapitola.

Vzdálená správa

Linux nabízí různé druhy vzdálené správy, například Telnet nebo SSH. K dispozici je také možnost správy přes vzdálenou plochu, nativně se o to stará protokol X, který je ovšem složitější na používání v síti. Spolehlivě lze použít i komerční software VNC.

Přizpůsobitelnost

Na rozdíl od rozšířeného konkurenčního operačního systému Windows se dá Linux jednoduše přizpůsobit účelům, pro které má sloužit, už při instalaci. Není tedy problém vyřadit nepotřebné aplikace a služby a danou distribuci nainstalovat přímo podle daných požadavků.

Linuxový router a hardware

Linux podporuje širokou škálu hardwaru. Nejčastěji najdeme podporu pro platformu IA-32. Ovšem spousta distribucí podporuje i další platformy, například DEC (firmy Compaq, pod křídly HP), AMD64, Itanium a jiné. Jako nevýhodu platformy IA-32 můžeme zmínit to, že její procesory dokáží adresovat maximálně 4GB RAM paměti. Ovšem pro router, který by měl sloužit pro účely domácnosti či malé firmy, takové množství paměti ani nebude zapotřebí.

Všechny požadavky na hardware se logicky odvíjejí od služeb, které se na daném stroji budou provozovat. Například již výše zmíněný DHCP server rozhodně nebude potřebovat 64 bitový procesor, vystačí si s procesorem třídy 80486. A vzhledem k tomu, že router budeme implementovat do prostředí malé sítě, tak se i se slabším procesorem lehce smíříme. Co se týče ostatních částí, diskový prostor stačí i do 2 GB, síťové adaptéry 100MB. Ani výkonnost grafické karty není nutná. Plně postačí integrovaná grafická karta, případně nějaká starší. Je však důležité, aby byla podporovaná.

Z výše uvedených hardwarových charakteristik je patrné, že Linux v roli routeru nebude potřebovat žádný silný stroj. Dokonce stačí nějaký funkční vyřazený počítač – velké firmy se takto poměrně často zbavují nedostačujícího vybavení. Pro práci byl zvolen následující hardware: Celeron 500 Mhz (tedy platforma IA-32), 128 SD-RAM (SDR), Voodoo 3000.

Výběr vhodné distribuce

U Linuxu, na rozdíl od jiných systémů, není jen jedna instalovatelná verze. Vzhledem k myšlence projektu GNU/Linux má každý právo vzít jádro systému a postavit si kolem něj vlastní software. Výsledek se pak nazývá distribuce. Vzhledem k povaze GNU/Linuxu několik hlavních distribucí založilo projekt LSB (Linux Base Standartisation), který se snaží standardizovat interní struktury linuxových operačních systémů. LSB vychází z otevřeného standartu POSIX.

Linuxových distribucí je mnoho. Od distribucí velmi malých, zpravidla máloúčelových, až po robustní distribuce schopné fungovat jako plnohodnotný desktop nebo server. Existuje i několik komerčních distribucí – dodávaných s technickou podporou. Pro účely linuxového routeru postačí bezplatná distribuce, která poskytne všechny potřebné služby, s ohledem na možné budoucí rozšíření v případě zájmu uživatele.

Všechny velké distribuce dnes nabízejí i instalaci pro server. Určitě se vyplatí volit nějakou větší distribuci, pokud to hardwarové požadavky dovolí, a to vzhledem k možné komunitní podpoře, kterou mnohdy menší distribuce neposkytnou.

Red Hat

Distribuce Linuxu Red Hat je vyvíjená společností Red Hat Inc. Od vydání Fedory verze 1 se společnost soustředí na vývoj distribuce Red Hat Enterprise, která je komerční. Red Hat je samozřejmě šířen pod licencí GNU GPL, takže je možné stáhnout zdarma zdrojové kódy. Placená je “jen“ podpora produktu. Řada firem je ovšem ochotna za tuto podporu platit, protože společnost Red Hat Inc. ručí za nápravu vad a garantuje technickou podporu, včetně včasné opravy. Z této distribuce vzešel známý balíčkovací formát RPM (RPM Package Manager), což je systém, který umožňuje daný program nainstalovat, případně odinstalovat, a někdy také dovoluje automatické stažení

z internetu. Další výhodou Red Hatu je poměrně široká škála GUI nástrojů pro správu. Hlavní platformou Red Hatu je IA – 32, podpora ovšem existuje i pro AMD64, Itanium a IBM servery. Starší verze Red Hatu podporovaly i platformu Alpha.

Fedora Core

Fedora Core představuje volně šiřitelnou komunitní distribuci odvozenou od Red Hatu. Fedora Core podporuje platformu IA-32 a AMD64.

Debian

Debian je jedna z mála distribucí, která je vyvíjena komunitně lidmi z celého světa. Debian má tři hlavní větve, do kterých se řadí software podle míry otestování. Jejich názvy jsou stable, testing a unstable. Stable větev nabízí plně otestovaný, funkční software zbavený všech chyb a řádně záplatovaný. Testing větev je určena převážně k testování. Unstable větev je používána vývojáři. Nevýhodou Debianu je pak možná zastaralost jednotlivých softwarových součástí stable verze, vzhledem k délce průchodu celým procesem unstable->testing->stable.

Často se o Debianu mluví jako o jednom z nejlépe přizpůsobitelných systémů. Co se týče balíčkovacích systémů, Debian používá nástroj Advanced Package Tool (APT), který umožňuje aktualizace přes internet. Tento systém se snaží řešit také závislosti mezi programy (ptá se, zda ma zaktualizovat souběžně s jedním programem i druhý apod.), nebo odstraňovat zastaralé komponenty. Debian je dostupný pro platformy IA-32, Alpha, IA-64…

Gentoo

Je další distribucí vyvíjenou výhradně komunitou. Gentoo je tzv. zdrojová distribuce, což znamená, že uživatel si může danou distribuci zkompilovat sám s potřebným programovým vybavením. Jako balíčkovací systém je zde použit Portage, který umožňuje překlad programů a následné přizpůsobení procesoru dostupným knihovnám apod. Výhodou je tedy maximální přizpůsobivost danému systému. Distribuce je dostupná pro IA-32, AMD64, SPARC….

SUSE

SUSE je distribuce spravovaná původně nezávislou německou firmou. V roce 2004 byla zařazena do portfolia firmy Novell. Tato firma je výhradním distributorem Linuxu s oficiální pobočkou v České Republice. Distribuce je placená a má masivní instalační a technickou podporu v češtině. Jako balíčkovací systém používá RPM. Za zmínku ještě stojí GUI konfigurační nástroje YaST2 a SaX2 pro správu hardwaru, nastavení systému a grafického rozhraní. SUSE Linux je určen především pro platformy IA-32 a AMD64.

Slackware

Slackware je nejstarší doposud aktivně vyvíjenou distribucí. Obsahuje pouze stabilní programy a klade důraz především na stabilitu, jednoduchost a konfigurovatelnost. Slackware používá vlastní balíčkovací systém pkgtools. Oficiálně je slackware pro platformu IA-32, ovšem existují i neoficiální porty pro platformy Aplha, SPARC, PowerPC, AMD64 a jiné.

Z výše uvedených charakteristik byla zvolena distribuce Slackware 12.0, která nabízí podporu pro IA-32 a tedy pro zvolenou konfiguraci hardwaru. Dále splňuje požadavky na slabší hardware, spolehlivost a přizpůsobitelnost.

Instalace slackware 12.1 na router

Instalace Slackwaru 12.1 na vybraný stroj bude určitě nejjednodušší pomocí CD nebo DVD. Vše potřebné se dá bezplatně stáhnout na www.slackware.org/getslack v podobě vypálitelného iso formátu. Další možností je instalace přes síť pomocí PXE. Tento způsob instalace ale musí být podporován BIOSem (i síťovou kartou) a pokud se pro router použije starší stroj je pravděpodobné, že tato podpora v BIOSu přítomna nebude.

V BIOSu daného stroje je následně nutné navolit bootovací pořadí tak, aby prvním zařízením, ze kterého se bude bootovat, byla CD/DVD mechanika.

Po úspěšném nabootování je nutné vybrat jádro. Zvolení jádra závisí na konkrétním hardwaru. Jádro huge.s je určeno pro platformu x86, ze které vychází

pozdější IA-32. Vzhledem k hardwarovým specifikacím je zvoleno jádro hugesmp.s, které je i přednastavenou volbou. Následuje zvolení klávesnice, respektive rozmístění kláves a zalogování se pod uživatele root (bez hesla).

Zařízení

V Linuxu je vše virtuálně reprezentováno jako soubor. V systémové složce /dev každý soubor reprezentuje to konkrétní zařízení. Proto i pevný disk bude umístěn v tomto adresáři. Nalezení pojmenování daného pevného disku je důležité pro další fázi instalace. Příkazem *dmesg* | *more* je možné vypsat seznam zařízení, kde se ukáže, pod jakým označením je pevný disk. V závislosti na rozhraní (ATA/SCSI) a zapojení (master/slave) potom Linux daný disk pojmenuje. ATA rozhraní bývají pojmenována jako /dev/hdx (kde x představuje písmena, začínají písmenem a – první ATA kanál, master; b – první ATA kanál, slave atd.). Naproti tomu SCSI rozhraním Linux přiřazuje jména /dev/sdx (se stejným systémem jako u ATA rozhraní). SCSI rozhraní je dobré zmínit z toho důvodu, že v Linuxu jsou takto reprezentovány například i usb disky nebo SATA disky.

Vytváření diskových oddílů

Instalace Slackware Linuxu vyžaduje alespoň jeden linuxový diskový oddíl. K vytváření diskových oddílů slouží programy cfdisk a fdisk. Program fdisk umožňuje provádět složitější operace s diskem. Pro základní nastavení ale bohatě postačí program cfdisk, který je jednodušší na ovládání. A to hlavně díky rozhraní v podobě grafického menu. Použití cfdisku (případně i fdisku) je cfdisk /dev/<jméno_zařízení>. Pomocí příkazu *new* se dá vytvořit nový diskový oddíl (ať už primární nebo logický) a následně určit, jaký rozsah bude mít. Typ oddílu je nativně nastaven jako Linux, takže se nemusí specifikovat. Optimální možností je vytvoření swap oddílu. Rozsah se většinou specifikuje v násobcích paměti, při 128 MB RAM bude 512 MB swapu bohatě stačit. Samozřejmě záleží také na celkových možnostech disku. Ne každý starý disk si takový luxus může dovolit. Po vytvoření swap oddílu je nutné specifikovat tento oddíl navolením, pomocí příkazu *Type*, čísla 82. Pro zapsání změn a opuštění programu cfdisk slouží příkaz *Write*.

Setup

Samotné instalační prostředí se spouští příkazem *setup*. Před instalací je nutné provést pár nastavení. Předně je nutné zvolit, odkud se bude Linux instalovat. Vzhledem k prozatímnímu průběhu bude instalace probíhat z CD/DVD mechaniky. Tu je nejlepší specifikovat manuálně, zvolením přesné lokace zařízení. K tomu opět poslouží příkaz *dmesg*. Následně se musí přiřadit souborové systémy k jednotlivým diskovým oddílům. Pro oddíl s operačním systémem Linux je dobré volit ext3 nebo reiserfs vzhledem k tomu, že umožňují žurnálování na rozdíl od ext2. Pokud jde o swap, instalátor sám provede několik příkazů po upřesnění daného oddílu. Dále je nutné zvolit software z připravených softwarových balíků, který se bude instalovat. Na routeru určitě nebudeme potřebovat grafická prostředí a hry. Následují už jen triviální nastavení, co se týče obtížnosti, není tím myšlena důležitost (například root heslo).

Intranetové služby

Směrování

Směrování (angl. routing) je proces, kde jsou jednotlivé IP-datagramy předávány z jednoho rozhraní do jiného rozhraní. Směrovač (router) se musí rozhodnout, do kterého rozhraní odešle IP - datagram, který obdržel. K tomu mu pomáhá směrovací (routovací) tabulka, kde jsou uvedeny důležité záznamy ohledně sítě, síťové masky, daného síťového rozhraní atd.

Rozhodování

V případě, že přijde na router IP-datagram adresovaný nějaké cílové stanici, router se musí rozhodnout jakému rozhraní ho předat. To se děje pomocí výše zmíněné směrovací tabulky. Router projde řádek po řádku směrovací tabulku, vezme síťovou masku a bit po bitu ji vynásobí s IP adresou příjemce v daném IP datagramu. Výsledek násobení se porovnává s prvním sloupcem směrovací tabulky. Pokud router najde více výsledků, které se shodují, vybírá si na základě metriky. Jestliže nastane situace, že router nenajde odpovídající shodu, posílá datagram implicitním směrem.

Směrovací tabulka se v Linuxu staticky plní, například při použití příkazu *ifconfig*, kterým se nastavuje síťové rozhraní (adresa, maska sítě…), nebo použitím příkazu *route*.

DHCP

DHCP (Dynamic Host Configuration Protocol) je z rodiny UDP protokolů typu client-server. Tento protokol umožňuje přiřazovat IP adresy z daného ip-poolu automaticky. IP adresa tedy nemusí být přesně specifikována pro jednotlivá rozhraní, ale danému klientovi je na požádání přidělena. Jakmile se spustí software DHCP na klientském počítači, vysílá klient do sítě požadavek na IP adresu. To se děje pomocí broadcastingu, kdy daný stroj rozesílá pakety do celé sítě (v IP datagramu je nastavena cílová adresa na 255.255.255.255). Daný server mu odpovídá a přiděluje adresu, pokud je vše v pořádku.

Nutno podotknout, že Slackware Linux obsahuje potřebný DHCP software od společnosti ISC, takže není nutné nic instalovat.

DHCP Server

DHCP server (tzv. DHCP) přiděluje adresu na žádost klienta. Vzhledem k tomu, že klient broadcastuje v rámci své sítě, je nutné, aby byl DHCP server přítomen v každé síti, respektive podsíti. Distribuce Slackware potřebným softwarem disponuje (pokud instalace proběhla dle popsaných instrukcí), konfigurační soubor je umístěn v /etc a jmenuje se dhcpd.conf. Samotná konfigurace se pak skládá se dvou částí – ze sady deklarací a ze sady parametrů. Deklarace popisují sítě, hostitelské počítače, rozsah adres který se dá přidělit jednotlivým klientům (tzv ip-pool), atd. Sada parametrů naproti tomu popisuje chování serveru a nastavení. Vnořené deklarace blíže specifikují další klienty, kteří patří pod danou deklaraci. Konfigurace souboru dhcpd.conf je možná pomocí textového editoru, například vim. Obecný konfigurační soubor bude mít následující strukturu:

```
Globální parametry;
Deklarace 1
        [parametry deklarace 1]
        [vnořená deklarace]

Deklarace 2
        [parametry deklarace 2]
        [vnořená deklarace]
```

Deklarace

Deklarace může sdružovat různé skupiny klientů. Vše se odvíjí od požadavků na danou síť. Vytváření umožňují tyto deklarace:

Group – tato deklarace aplikuje konkrétní skupinu parametrů na seznam klientů, podsítí nebo sdílených sítí.

```
group label
        [parametry]
```

[vnořená deklarace]

Label je jméno dané skupiny.

Host – deklarace host se používá pokud chceme konkrétnímu klientovi IP adresy danému klientovi.

host label
[parametry]
[vnořená deklarace]

Label symbolizuje daného klienta.

Shared–Network – sdružuje skupinu adres klientů se stejnou fyzickou sítí.

shared-network label
[parametry]
[vnořená deklarace]

Label v tomto případě je jméno sdílené sítě.

Subnet – tato deklarace se požívá při definování deklarací a parametrů pro danou podsíť.

subnet subnet-number **netmask** netmask
[parametry]
[vnořená deklarace]

subnet-number reprezentuje síť, jejímž klientům budou přidělovány IP adresy. netmask je potom maska sítě dané podsítě.

Range – specifikuje rozsah adres, které se budou přidělovat jednotlivým klientům.

range [dynamic-bootp] starting-adress [ending-adress]

dynamic-bootp je volitelné nastavení, které sdělí serveru, že rozsah přiřazovaných adres je určený pro protokol bootp (protokol, ze kterého je DHCP

defakto odvozeno). Starting-adress a ending-adress představují hranice bloku adres, ze kterého se bude přiřazovat.

Parametry

Parametry (nebo také volby) blíže specifikují danou skupinu klientů. Patří mezi ně:

authoritative – označí síť jako autoritativní. Autoritativní síť se potom ke klientům chová tak, že na jejich špatný DHCPREQUEST (způsobený vyžádáním adresy, která neodpovídá dané podsíti, například při nastavené adrese jinou podsítí) pošle DHCPNAK odpověď, což navrátí klienta zpět do stavu INIT. Při nenastavení se tedy tomu tak nestane.

authorative;
not authorative;

max-lease-time – dhcp klient může požádat o prodloužení lhůty, po kterou má poskytnutou adresu. Server potom lhůtu prodlouží, pokud klient nepřekročil maximální dobu.

max-lease-time seconds;

default-lease-time – výchozí doba trvání pronájmu adresy.

default-lease-time seconds;

filename – obsahuje cestu k bootovacímu souboru. Tuto volbu používají bezdiskové stanice. Často bývá kombinováno s parametrem next-server.

filename filepath;

fixed-adress – parametr deklarace host. Přidělí konkrétní IP adresu, případně skupinu adres, konkrétnímu klientovi.

fixed-adress adress [, adress .];

hardware – tento parametr nastaví hardwarovou adresu (MAC adresa), která je charakterizována hexadecimálním oktetem, a typ linkové vrstvy. Používá se současně s parametrem fixed-adress.

hardware [ethernet | token-ring] hardware – adress;

option host-name – název klientského počítače.

option host-name name;

option domain-name – název domény klienta.

option domain-name name;

option routers – adresa routeru.

option routers adress;

option domain-name-servers – název DNS serveru, případně jeho IP adresy.

option domain-name-servers [adress | IP];

ddns-update-style – tento parametr říká jakým způsobem je řešena synchronizace DNS a DHCP

ddns-update-style [ad-hoc|inerim|none];

use-hosts-decl-names – tento parametr přiřadí danému klientovi hostname, jaké se používá pro deklaraci host.

use-hosts-decl-names [true | false];

DHCP klient

DHCP klient používá také konfigurační soubor, ovšem není nutné ho konfigurovat v případě jednoho síťového rozhraní na klientské stanici. Pakliže je klientem stanice s operačním systémem Linux, mělo by vše proběhnout bez problémů a klientský počítač by měl obdržet IP adresu dle nastavení. V případě stanice s operačním systémem Windows je to trochu složitější.

Potencionální problém se týká všesměrového (broadcast) vysílání. Vysílání typu broadcast se dělí na globální a lokální. Globální broadcast používá adresu 255.255.255.255, lokální broadcast je naproti tomu určen jen klientům dané podsítě. Klient stanice Windows očekává odpověď pomocí globálního broadcastu. Ovšem linuxový stroj převede globální broadcast na lokální a tím pádem dochází k nepřidělení adresy. Celý tento problém se dá poměrně jednoduše napravit pomocí příkazu:
route add –host 255.255.255.255 dev jméno_sitoveho_rozhrani
Jméno síťového rozhraní se pak upraví podle toho, do jaké sítě přiděluje DHCP server adresy.

DNS

DNS neboli name server slouží k překladu doménových jmen na IP adresy a naopak. Důvodem pro překlad je samozřejmě lepší zapamatovatelnost názvů než čísel. Počátek prvních řešení převodu IP adres na jména sahá do 70. let 20. století, kdy vznikal TCP/IP protokol. Překlad se tehdy řešil pomocí souboru hosts, který se musel udržovat neustále aktuální (dělo se tak pomocí FTP). Začátkem 80. let však stoupá počet počítačů nad únosnou mez a udržování jednoho souboru je de facto nemožné. Řešení přichází v podobě distribuovaného systému, kde každá část sítě udržuje informace o vlastních prvcích.

DNS Resolver

Resolver neboli česky rozkladač představuje v DNS systému klientskou část, přestože se nejedná o konkrétní aplikaci. Umožňuje evidování DNS serverů, na které budou dotazy o překlad adres směrovány. Takže pokud aplikace potřebuje, zadá operačnímu systému požadavek na překlad adresy a počká si na odpověď, nemusí tedy sama implementovat daný protokol.

V Slackware linuxu se soubor nachází v /etc/resolv.conf s jednoduchou syntaxí v podobě:

```
search nazev_domeny
nameserver IP_adresa
```

Počet name serverů není v konfiguračním souboru nijak omezen. Více serverů se specifikuje k záložním účelům – používá se server na prvním místě, v případě jeho nedostupnosti se použije server na druhém místě atd.

Typy DNS serverů

Resolver slouží pouze pro pokládání dotazu cílovému serveru. Veškerý překlad a tedy celý systém leží na DNS serverech. Podle funkce a postavení v hierarchii DNS můžeme rozlišit několik typů DNS serverů:

- Primární DNS server: jak již název napovídá jde o hlavní DNS server, obsahuje veškeré informace o doménách náležících dané síti.
- Sekundární DNS server: záložní server, který si automaticky udržuje stejné informace, které obsahuje primární DNS server. Používá se v případě výpadku primárního DNS serveru.
- DNS cache server: slouží k vyřizování dotazů. Zároveň si pamatuje odpovědi, díky čemuž může výrazně přispět k urychlení provozu na síti.

Konfigurace DNS serveru

Pro DNS server použijeme software BIND. Jedná se o jeden z nejrozšířenějších serverů pro DNS. Slackware Linux software BIND samozřejmě obsahuje, takže není nutné nic instalovat. Konfigurace samotná se potom provádí v souboru /etc/named.conf.

Zóny

Do tohoto souboru je nutné nejdříve přidat řádek, který bude odkazovat na konkrétní konfigurační soubory. Jednoduše tedy lze přidat:

```
directory “/var/named”
```

Dále je nutné do souboru etc/named.conf zanést zóny. Zóna je vlastně subdoménou dané domény. Syntaxe zóny bude následující:

```
zone jmeno_zony {
        type master;
        file cesta;
        };
```

Pro překlad názvu na IP adresu a pro překlad IP adresy na název musí existovat unikátní zóny. Překlad IP adresy na název se často nazývá reverzním překladem. Pro vytvoření této zóny je nutné zanést název zóny in-addr.arpa a ostatní sounáležitosti do /etc/named.conf. Celé jméno reverzní zóny se skládá z výše uvedeného názvu a prvních třech bajtů IP adresy napsaných pozpátku (například pro adresu 192.168.1.1 to bude 1.168.192.in-addr.arpa).

Dále je možné ještě doplnit sekundární zónu pro případ záložního DNS serveru. Syntaxe je v takovém případě následující:

```
zone jmeno_zony {
        type slave;
        masters seznam_IP_adres;
        };
```

Povinnou součástí konfigurace je nastavení cache zóny. Nejdříve se přidá zóna pro cache:

```
zone “.” {
        type hint;
        file “named.ca“;
        };
```

Typ hint identifikuje zónu jako cache. Soubor named.ca je součástí balíku BIND, takže nebude potřeba nic konfigurovat. Druhým záznamem je:

```
zone “0.0.127.in-addr.arpa” {
```

```
type master;
file “named.local“;
};
```

Tato zóna umožní rekurzivní prohledávání stromové struktury DNS.

Typy záznamů DNS

Konkrétní databázové soubory nesoucí záznamy mají určitý formát. Typy záznamů, které můžeme použít jsou SOA, NS, A, CNAME, PTR, MX, TXT a RP.

SOA – znamená Starting of Authority a začíná definice dalších DNS záznamů. Syntaxe:

```
IN SOA name_server.domena.cz. host_master.domena.cz. (
        #######                 ; sériové číslo
        #######                 ; obnovovací frekvence v sekundách
        #######                 ; opakované pokusy v sekundách
        #######                 ; vypršení v sekundách
        #######     )           ;minimum v sekundách
```

name_server.domena.cz. představuje hostitelský počítač pro konfigurační soubor.

host_master.domena.cz. je elektronická adresa správce domény.

První řádek v závorce udává číslo, které symbolizuje poslední aktualizace, proto bývá ve formátu YYYYMMDD. Druhý řádek určuje, za jak dlouho se sekundární server zeptá primárního, zda nedošlo ke změně. Třetí řádek udává, kdy se sekundární server bude snažit kontaktovat primární v případě, že předtím neuspěl. Poslední řádek udává, kdy sekundární server přestane kontaktovat primární, pokud se mu to nedaří.

NS – záznam typu NS (name server) slouží k určení jmenných serverů spravujících informace o dané zóně.

```
IN NS name_server.domena.cz.
```

IN NS name_server.domena.cz.

A – slouží k převodu názvu stanice na IP adresu.

název_stanice IN A IP_adresa

PTR - ve slovním významu opak A. Slouží k překladu IP adresy na název.

IP_adresa IN PTR název_stanice

MX – neboli Mail Exchanger. Tento typ udává informace o serveru, který slouží k doručování pošty.

nazev_domeny.cz. váha nazev_stanice

CNAME – umožňuje vytváření aliasů pro název jakékoliv stanice. Často se toto označení užívá při změně názvu poskytované služby, v okamžiku, kdy zákazníci ještě nejsou na nový název produktu zvyklí.

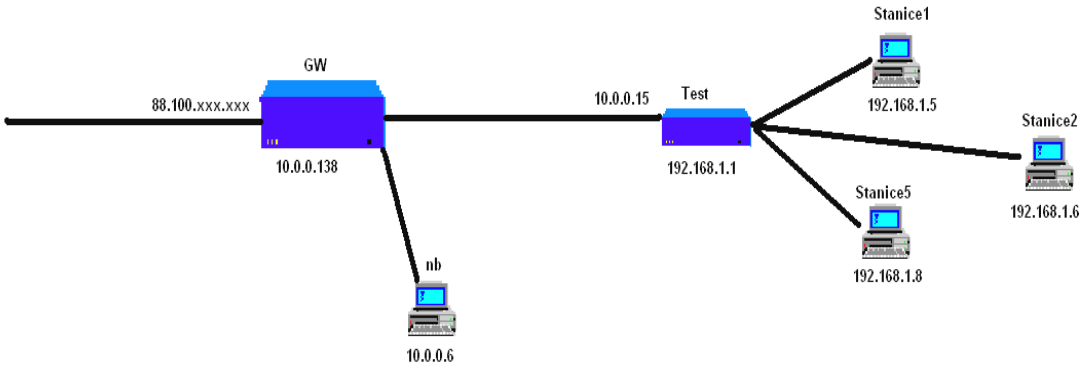
novy_nazev_stanice IN CNAME stary_nazev_stanice

Nastavení routeru

Spouštěcí scripty

Vzhledem k tomu, že se v průběhu konfigurace upravuje poměrně značná část nastavení a Linux by je po restartu znovu nezavedl, je nutné tato nastavení přidat do spouštěcích scriptů. Veškerá nastavení, která se nebudou týkat samostatných konfiguračních souborů daného softwaru a služby, se tedy zapíší do scriptu rc.local uloženém v /etc/rc.d/rc.local.

Popis sítě



Ob.r č.1 Topologie sítě.

Testovací router se nachází uvnitř jiné sítě, konkrétně jde o 10.0.0.0/24. Jedno síťové rozhraní bude dostávat ip adresu z routeru, který je označen jako GW, druhé rozhraní pak poslouží pro subnet 192.168.1.1, které bude poskytovat i ostatní služby.

Nastavení sítě

Samotné routování se zapne pomocí příkazu *echo*, kterým vložíme parametr 1 do souboru /proc/sys/net/ipv4/ip_forward.

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

Pro správné routování je nejdříve nutné nastavit rozhraní. Protože první rozhraní (eth0) komunikuje se sítí 10.0.0.0, na které funguje DHCP server, je možné potřebné informace pro první rozhraní získat automaticky. Provede se to jednoduše spuštěním DHCP klienta na rozhraní eth0. Konkrétně tedy příkazem:

```
dhcpcd eth0
```

DHCP klient má vlastní konfigurační soubor, ale není nutné jej upravovat pro korektní fungování. Tím se tedy získají první důležité údaje jako je ip adresa, síťová maska apod.

Pro druhé rozhraní (eth1) se tyto údaje nastaví ručně. V Linuxu pro to slouží příkaz *ifconfig*. Syntaxe vypadá takto:

```
ifconfig dev ip netmask nmask volby
```

Konkrétně to tedy bude:

```
ifconfig eth1 192.168.1.1 netmask 255.255.255.0
```

Pro kontrolu lze zadat samotný příkaz *ifconfig*, který vypíše seznam rozhraní a jejich nastavení.

Konfigurace DHCP Serveru

S uvedenou teorií je poměrně jednoduché napsat konfigurační skript, který uvede DHCP server do provozu. Konfigurační soubor dhcp daemona se nachází v /etc/ a má jméno dhcpd.conf. Pomocí textového editoru (vim například) se dá napsat skript, který zajistí korektní přidělování adres.

```
authoritative;
option domain-name “tdomain.cz“;
option routers 192.168.1.1;
option domain-name-servers 192.168.1.1;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
default-lease-time 86400;
max-lease-time 604800;
subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.5 192.168.1.30;
}
group {
        use-hostdecl-names true;
        host stanice1 {
                hardware ethernet 00:01:4a:06:6a:4a;
```

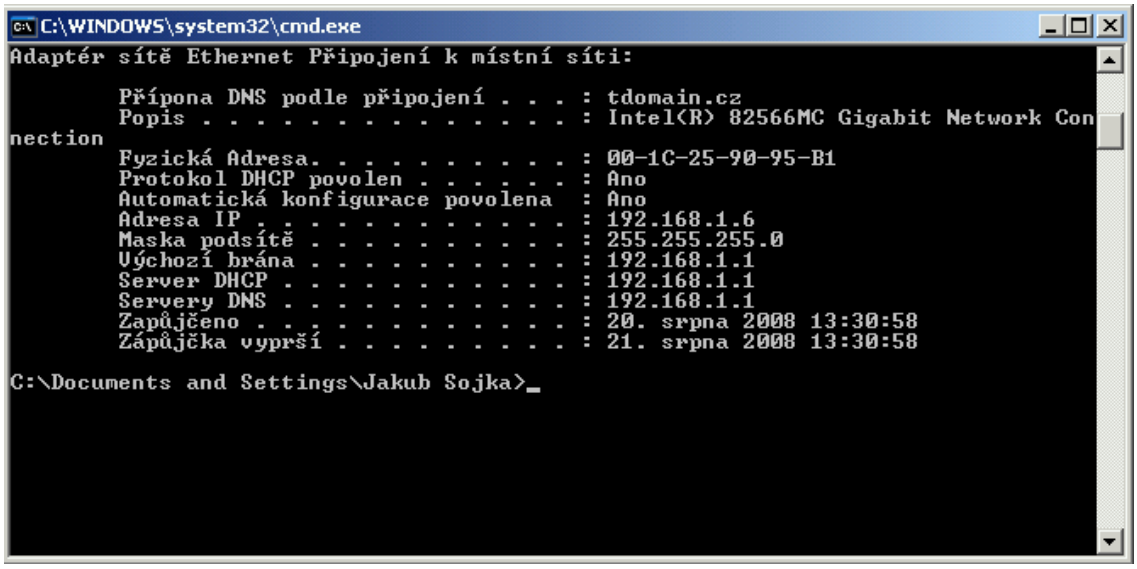
```
        fixed-address 192.168.1.5;
                }
host stanice2 {
        hardware ethernet 00:1c:25:90:95:b1;
        fixed-address 192.168.1.6;
                }
host stanice5 {
        hardware ethernet 00:0e:2e:f3:41:c2;
        fixed-address 192.168.1.8;
                }

}
```

DHCP server v tomto případě poskytuje adresy v rozsahu 192.168.1.5 až 192.168.1.30. Adresy 192.168.1.5, 192.168.1.6, 192.168.1.8 jsou rezervovány pro počítače s názvem stanice1, stanice2 a stanice5, což znamená, že je DHCP server nebude poskytovat jiným rozhraním.

Samotný DHCP server se pouští příkazem:

```
dhcpd
```

V případě spojení se stanicí s operačním systémem Linux je nutné pro přidělení adresy pustit DHCP klienta (pokud není implicitně zapnut). Stanice Windows mají většinou DHCP klienta zapnutého, tudíž není potřeba nic pouštět. V případě dodržení popsaného postupu je možné obdržet takovýto výsledek (v tomto případě šlo o výsledek pro počítač s názvem stanice2):



Obr. č. 2 Automatické přidělení ip adresy.

Konfigurace DNS

Pro službu DNS je nutné nejdříve upravit zóny v souboru /etc/named.conf. Na testovacím routeru se původní obsah souboru named.conf ponechal a došlo k přidání následujících řádků:

```
zone “tdomain.cz” IN {
        type master;
        file “tdomain.db“;
        };
```

A k ní reverzní zóna:

```
zone “1.168.192.in-addr.arpa“ IN {
        type master;
        file “tdomain.rev”;
        };
```

Oba konfigurační soubory jsou uloženy ve /var/named. Cesta k souborům je specifikována v /etc/named.conf pomocí příkazu *directory*, jak již bylo výše zmíněno. Nyní se musí vytvořit databázové soubory. Neexistuje žádný standart, jak by měly být tyto soubory správně psány, avšak v průběhu let se vžilo pár pravidel, která jsou v souvislosti se psaním těchto konfiguračních souborů dodržována. Každý ze souborů musí začínat záznamem $TTL, tím BIND server pozná, jaká je doba platnosti

jednotlivých záznamů. Následně musí následovat SOA záznam a alespoň jeden NS záznam. Vše ostatní je nepovinné.

Konfigurační soubory pro testovací router pod operačním systémem Slackware Linux vypadají následovně:

Soubor /var/named/tdomain.sb :

```
$TTL 86400
@       IN      SOA   tdomain.cz. root.tdomain.cz. (
        20080117
        10800
        1800
        1209600
        604800    )
        IN      NS    tdomain.cz.
        test    IN      A       192.168.1.1
        stanice1        IN      A       192.168.1.5
        stanice2        IN      A       192.168.1.6
        stanice5        IN      A       192.168.1.8
```

Soubor /var/named/tdomain.rev :

```
$TTL 86400
@       IN      SOA   1.168.192.in-addr.arpa. root.tdomain.cz. (
        20080117
        10800
        1800
        1209600
        604800    )
        IN       NS     tdomain.cz.
1       IN      PTR   test.tdomain.cz.
5       IN      PTR   stanice1.tdomain.cz.
6       IN      PTR   stanice2.tdomain.cz.
8       IN      PTR   stanice5.tdomain.cz.
```

V tomto případě je vhodné ještě před samotným spuštěním DNS serveru zkontrolovat syntaxi všech konfiguračních souborů. V případě souboru named.conf se to provede jednoduše pomocí příkazu:

named-checkconf

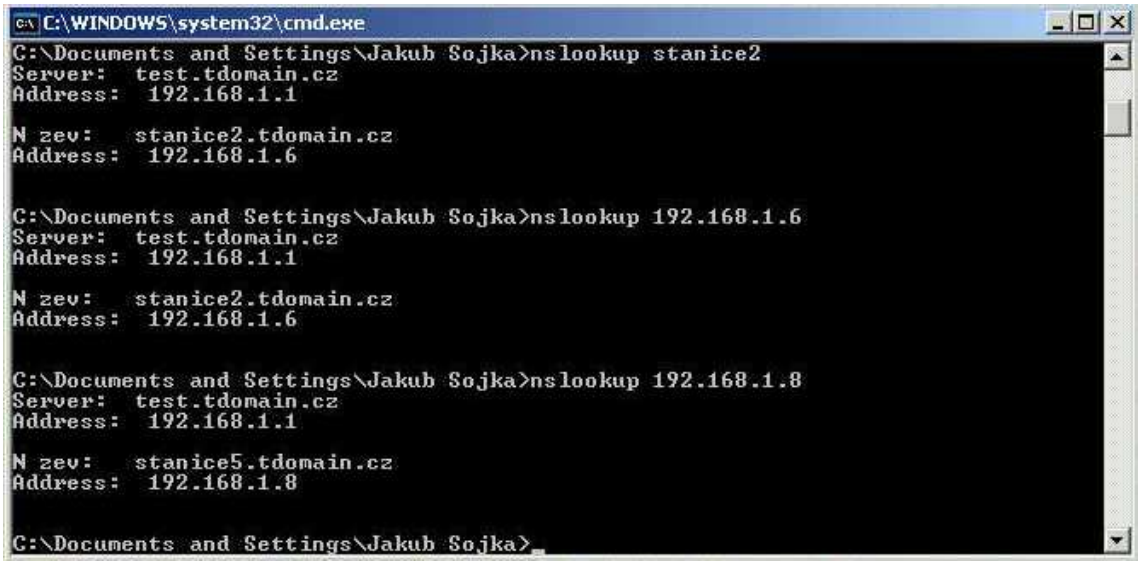
V případě databázových souborů se použije příkaz se syntaxí:

named-checkzone nazev_zony cesta_k_souboru

Pokud daný příkaz vrátí nápis OK, mělo by být vše v pořádku a DNS server je možné pustit příkazem:

named

Správnost nastavení je možné ověřit z některé stanice v síti 192.168.1.0. V případě stanice s OS Windows se použije příkaz nslookup. Daný výsledek může vypadat například takto:



Obr. č. 3 Překlad jména/ip adresy.

Obrázek ukazuje i reverzní překlad – z IP adresy na jméno.

Databázové soubory představují to nejpodstatnější z celého nastavování BIND serveru. Zpětný překlad adres je vyžadován některými službami (Sendmail…), a proto je vždy dobré mít tuto konfiguraci připravenou. Vzhledem k možnosti výskytu různých dedikovaných strojů (WWW server apod.) v síti je konfigurace DNS serveru esenciální.

Zabezpečení

Bezpečnostní politika serveru, respektive routeru, se skládá z několika podmínek, které je nutné splnit. Získat absolutně bezpečný systém je téměř nemožné, protože případní útočníci se neustále zdokonalují, stejně jako jejich nástroje. Samotné zabezpečení by vydalo (a ono už také vydalo) na bezpočet samostatných knih. Vzhledem k povaze práce je dobré zmínit alespoň základní pravidla zabezpečení, firewall a možné druhy síťových útoků, které by takto nakonfigurovaný router mohly ohrozit. Výše zmíněná konfigurace nahrává samotnému zabezpečení, protože neobsahuje velký počet služeb.

Fyzické zajištění routeru

Bezpečnost systému není založena jen na vhodné konfiguraci softwaru, případně bezpečnostních nástrojů, ale velkou roli zde hraje i fyzické zajištění hardwaru.

Pokud útočník pronikne až přímo ke stroji, existuje jen málo možností jak mu zamezit průniku do systému. Proto první věcí, kterou je nutno zajistit, je bezpečná místnost, nejlépe bez přístupu každého ze zaměstnanců. Další možností fyzického zabezpečení je použití uzamykatelné skříně.

Nebezpečí pro počítače nehrozí jenom od lidí, ale i od různých fyzikálních vlivů. Mezi ty nejhorší patří elektřina, teplota, oheň apod.

Elektřina je pro zajištění chodu routeru esenciální. Zdroje jsou vyvíjeny tak, aby dokázali vydržet výkyvy v napětí, ale nejsou nezničitelné. V případě nasazení routeru v prostředí firmy, a s ohledem na běžící služby, je výhodné zajistit stroj UPS zdrojem. UPS zdroj udrží v provozu systém tak dlouho, aby se byl schopen korektně vypnout.

Teplota není zase až takový problém, pokud se jedná o malý router. Určitě ale ničemu neuškodí uložení gateway v dobře klimatizované a suché místnosti.

Zatímco elektřina a teplota nemusí mít fatální následky pokud se něco pokazí, pro oheň to neplatí. Každá místnost, kde je sofistikovaná výpočetní technika, by měla být proto vybavena hasicím přístrojem. Na elektroniku je nutné používat práškové hasicí přístroje.

Vzdálený přístup

Pokud je potřeba na routeru něco nastavit, případně se dostat někam uvnitř sítě, je dobré mít v provozu nějakou metodu vzdáleného přístupu. Jednou z možností je například služba Telnet.

Telnet

Pokud se do sítě připojujeme přes internet, je Telnet dírou v zabezpečení. Telnet totiž přenáší všechna data nezašifrovaná a umožňuje je tak komukoliv, kdo poslouchá na správných linkách, vidět. To platí i pro hesla! Celá spojení lze tedy jednoduše přesměrovat na jiný stroj a pokračovat v ovládání z něj.

Telnet bývá často nasazen uvnitř sítě pro rychlou konfiguraci serverů.

SSH

SSH je další protokol umožňující vzdálenou správu, ovšem jeho výhodou oproti telnetu je šifrovaný přenos dat. Díky implementovanému systému klíčů si mohou obě strany ověřit kdo s nimi komunikuje.

Klíče a otisky

SSH používá asymetrické šifrování, které obsahuje systém párových klíčů. Klíč je v tomto případě reprezentován jako nějaká data, pomocí kterých počítač se zprávou pracuje. Klíče mají několik zajímavých vlastností: jeden klíč se vždy použije pro zašifrování zprávy, druhý k jejímu dešifrování, klíče jsou od sebe neodvoditelné, klíče nesouvisí s danou zprávou, tudíž je neodvodíme ani z ní.

Otisky se používají na ověření správnosti klíče. Veřejné klíče jsou dlouhé řetězce písmen, čísel a znaků, a proto by je bylo náročné ověřovat v takové podobě. SSH ale dokáže z tohoto řetězce vygenerovat unikátní matematický otisk, který pak předkládá uživateli. K překladu tohoto otisku se používají vyspělé hashovací funkce, které zajistí, že i při nepatrné změně klíče se změní jeho otisk.

SSH klient

Součástí OpenSSH je i client. Syntaxe příkazu je následující:

ssh -l uzivatel nazev_stroje/ip_adresa

Poměrně důležitý je přepínač –l, který zajistí použití jiného přihlašovacího jména, než toho pod kterým je uživatel zrovna přihlášený.

Klienti s operačním systémem Windows mohou využít známý program PuTTY.

Komunikace mezi klientem a serverem pak prochází několika fázemi: klient se se serverem propojí na portu 22 (pokud je využíván pro komunikaci defaultní port). Server předá klientovi veřejný šifrovací klíč. Klient klíč ověří a vygeneruje klíč pro komunikaci. Tento klíč pak zašifruje veřejným klíčem a odešle serveru. Následuje bezpečná komunikace mezi serverem a klientem.

Firewall

V Linuxu zajišťuje firewall část jádra, která se nazývá iptables. Tato služba neobsluhuje jenom firewall, ale umožňuje defakto jakoukoliv manipulaci s pakety.

Tabulky a řetězce

Každý paket prochází systémem řetězců, které jsou součástí tabulek.

Základní tabulky, které můžeme najít, jsou tři:

- nat – tato tabulka se používá pro překlad adres (NAT = Network Adress Translator). Překlad se používá, pokud je pro vnější rozhraní dostupná jen jedna IP adresa a uvnitř sítě se nachází větší množství klientů.
- filter – řeší filtrování paketů, jejich zahazování nebo logování.
- mangle - tato tabulka umožňuje modifikaci paketu – nastavit TTL, nastavení TOS pole v IP zahlaví apod.

Nejčastěji se používá tabulka filter. Mezi základní řetězce tabulky filter patří:

- INPUT – do tohoto řetězce vstupují všechny příchozí pakety.
- OUTPUT – do tohoto řetězce vstupují všechny odchozí pakety.
- FORWARD – do tohoto řetězce vstupují všechny pakety, které se přeposílají (typické pro router).

Příkaz iptables

Iptables manipulují s řetězci pomocí specifikovaných pravidel. Pravidlo obsahuje podmínky, které musí daný paket splňovat a cíl, který se má vykonat. Nesplňuje-li podmínky konkrétního pravidla, aplikuje se na něj pravidlo další, dokud se nedojde k implicitnímu cíli řetězce. Syntaxe iptables je následující:

```
iptables [tabulka] [akce] [retezec] [pravidla] [cil]
```

Akce může mít několik parametrů:

- -A – na konec řetězce se přidá nové pravidlo.
- -I – na začátek řetězce se přidá nové pravidlo.
- -D – smazání pravidla podle čísla.
- -L – výpis pravidel daného řetězce.
- -P – zadání hlavního pravidla.

Důležitou součástí příkazu *iptables* jsou pravidla, podle kterých filtruje. Mezi nejběžnější pravidla patří:

- -s – zdrojová adresa paketu.
- -d – cílová adresa paketu
- -i – vstupní rozhraní pro příchozí paket.
- -o výstupní rozhraní pro odchozí paket.
- --sport – zdrojový port paketu.
- --dport – cílový port paketu.

Poslední věc, která se musí nastavit je to, co se s paketem stane. Slouží k tomu cíl symbolizovaný přepínačem –j. Cílem bude buď nějaký jiný řetězec do kterého paket spadne nebo nějaká akce:

- ACCEPT: Přijme paket. Paket tedy projde filtrem.
- DENY: Zamítne paket a odešle o tom zprávu stroji, který paket vyslal.
- DROP: Zamítnutí paketu bez odeslání zprávy zdroji.

Zabezpečení routeru

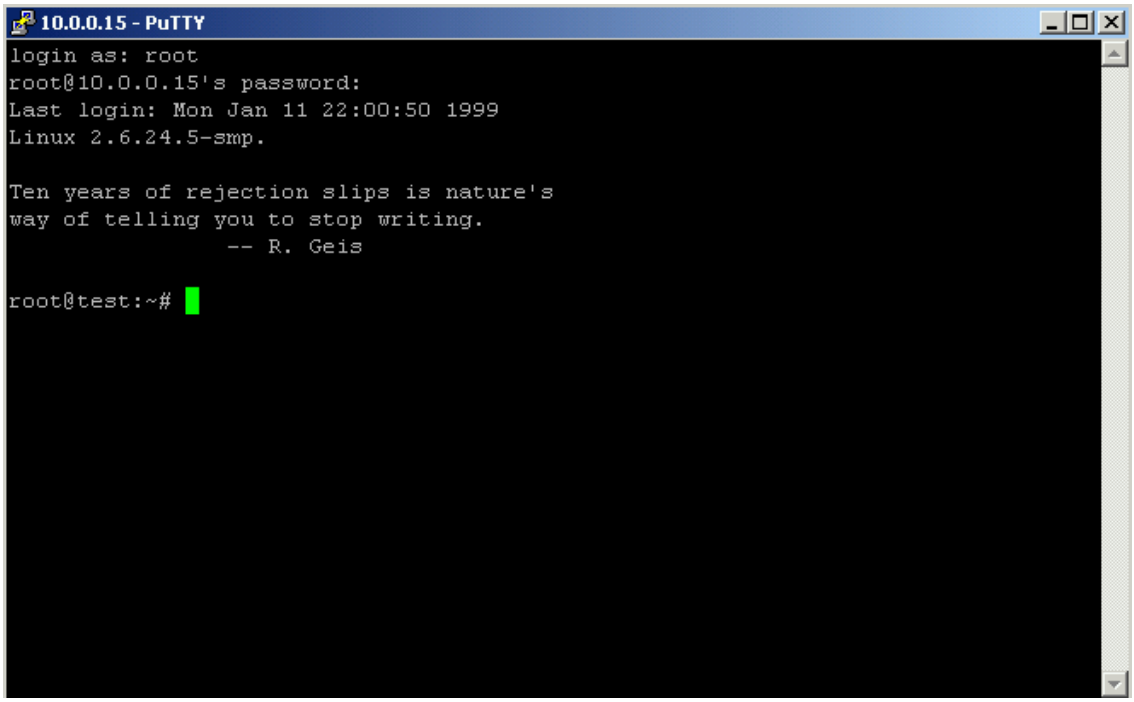
Konfigurace SSH serveru

Běh serveru zajišťuje daemon sshd, který se nachází v /usr/sbin/. Konfigurační soubor daemon se nalézá v /etc/ssh/sshd_config. Struktura tohoto konfiguračního souboru je poměrně jednoduchá. Na každém řádku se nalézá nějaký parametr, který je možné nastavit. Standardně konfigurační soubor nabízí mnoho voleb, které jsou ovšem zakomentovány. Pro použití se tedy musí určité řádky odkomentovat (odtraněním # ze začátku řádku):

```
Port 22
Protocol 2,1
IdentityFile /etc/.ssh/identity
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
LoginGraceTime 2m
PermitRootLogin no
RSSAuthentification yes
PubKeyAuthentification yes
AuthirizedKeysFile  .ssh/authorized_keys
PasswordAuthentification no
PermitEmptyPasswords no
X11Forwarding no
TCPKeepAlive yes
UsePrivilegeSeparation yes
Compression yes
UseDNS yes
MaxStartups 10
```

Po odkomentování těchto řádků je nutné pustit daemona. Pro jeho spuštění je nutné specifikovat celou cestu, tzn. /usr/sbin/sshd. Odkomentování výše umíněných

řádku v konfiguračním souboru ssh serveru stačí na jeho zprovoznění. Tato konfigurace se zkoušela ze sítě 10.0.0.0, pomocí programu PuTTY a vše proběhlo v pořádku:



Obr. č. 4 Test ssh serveru.

Pro spouštění ssh daemona po startu operačního systému je samozřejmě znovu nutné připsat cestu /usr/sbin/sshd do spouštěcích skriptů (/etc/rc.d/rc.local).

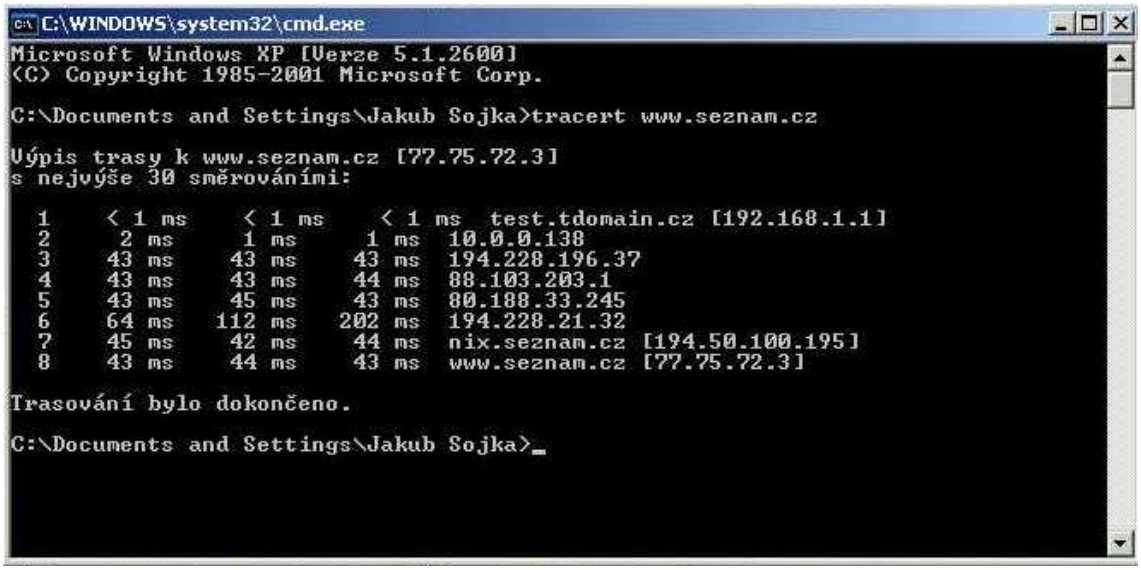
Nastavení iptables

Na konkrétně řešeném routeru nastavíme několik účinných pravidel, které napomohou zabezpečení. Veškerá pravidla je nutné zapsat do spouštěcích skriptů.

iptables –P INPUT DROP
iptables –A –i lo –j ACCEPT
iptables –A INPUT –i eth0 –dport 22 –j ACCEPT
iptables –t nat POSTROUTING –s 192.168.1.0/24 –o eth0 –j MASQUERADE

V případě jednoduché domácí sítě tato pravidla postačí. Samozřejmě je nutné přizpůsobovat firewall síti a službám které pod ní běží. První řádek se nastaví jako hlavní pravidlo a bude zamítat veškeré příchozí pakety. Druhý řádek povoluje vstup na loopback. Třetím pravidlem se povoluje nastavené ssh, tak aby bylo přístupné

z rozhraní, které míří ven. Čtvrté pravidlo je nutné pro správné fungování ip forwardu. Samozřejmě pokud má daný stroj od ISP určitý rozsah ip adres, které může použít, tak se to dá také v iptables nastavit a vše bude fungovat podle routovací tabulky (při předpokladu, že router přidělí jednotlivým počítačům ip adresy z dostupného rozsahu). Ovšem v případě jedné veřejné ip adresy je nutné nastavit maškarádu, což zajistí přepsání adresy z privátního rozsahu na veřejnou adresu a tedy odeslání paketů správným směrem. Zároveň to přispívá k zabezpečení sítě, protože maškarádovací tabulka se plní dynamicky na popud daného stroje v privátním rozsahu a nelze tedy k tomuto stroji přistoupit přímo. Nevýhodou pak může jedině být to, pokud máme uvnitř sítě nějaký dedikovaný stroj (například WWW server). V tomto případě se musí port na routeru, kam přijde daný paket, forwardovat na danou vnitřní ip adresu a port. Poslední pravidlo tedy zajistí správné forwardování paketů, o kterém se můžeme přesvědčit například příkazem tracert ze stanice s operačním systémem Windows:



Obr. č. 5 Forward paketů pomocí iptables.

Závěr

Technický rozvoj v posledních letech a hlavně rozmach internetu znamenají potřebu malých sítí. Routery v nich umístěné jako internetové brány často nenabízejí tak široký výběr služeb, jaký by mohly. S využitím vlastností operačního systému GNU/Linux je možné dosadit do této role vyřazený funkční počítač, který ji zdárně zastane. Současně se stoupající popularitou Linuxu už není nastavení podobné brány záležitostí jen pro zasvěcence. Nabídka distribucí a nástrojů je tak velká, že dokáže pokrýt jakýkoliv požadavek a díky vývoji grafických rozhraní se dá mnoho distribucí jednoduše konfigurovat.

V první části této práce se na základě osobní zkušenosti a poznatků z odborné literatury zkoumaly hlavní rysy operačního systému Linux. Podle popisu distribucí pak byla zvolena distribuce Slackware, která splňovala stanovené požadavky nejlépe. Popsána byla stěžejní část celého procesu implementace a sice instalace, která od dob dřívějších verzí zaznamenala mnohé pokroky.

Dále se práce zabývala nastavením sítě a služeb, kde se ukázalo, že není tak těžké vytvořit konfigurační skripty, které zajistí základní funkce většiny dnešních routerů a zároveň připravili router pro možné další služby. Veškerý software, který se použil byl získán zdarma pod licencí GNU GPL, což zajistilo nulové počáteční náklady za software. Jako hardware byl použit vyřazený stroj se kterým Linux neměl problém komunikovat.

Linux tedy jde poměrně lehce nasadit jako směrovač. Navíc jde o plnohodnotný operační systém, který se dá přizpůsobit budoucím požadavkům. Díky neustálé podpoře starších verzí a možností upgradu ani nezestárne, takže se uživatel nemusí bát, že by ho výrobce přestal podporovat.

Seznam obrázků

Seznam literatury

1. SHAH, Steve, SOYINKA, Wale. Administrace systému Linux.Vydání první. Praha:Grada 2007.ISBN 978-80-247-1694-7
2. DOSTÁLEK, Libor, KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. Vydání třetí. Brno: CP Books 2005. ISBN 80-7226-675-6
3. KRČMÁŘ, Petr. Linux - tipy a triky pro bezpečnost. Praha: Grada 2004. ISBN 80-247-0812-4
4. VESELÝ,Arnošt. Operační systémy II. Vydání první. Praha: Česká Zemědělská Univerzita, Provozně Ekonomická Fakulta 2001.ISBN 80-213-1553-9
5. KRČMÁŘ,Petr. Linux – postavte si počítačovou síť.Vydání první. Praha:Grada 2008. ISBN 978-80-247-1290-1
6. SMITH,Roderick W. Linux ve světě Windows. Vydání první. Praha: Grada 2006.ISBN 80-247-1470-1
7. VONDRÁČEK,Jan.Linux jako internetová gateway (4). [online] < http://www.root.cz/clanky/linux-jako-internetova-gateway-4/ >