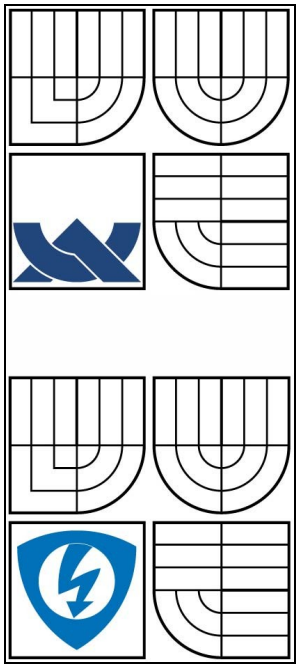


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ŘEŠENÍ HOSTINGOVÝCH SLUŽEB NA OPENSOURCE PLATFORMÁCH

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE Bc. ONDŘEJ MATĚJÍČEK
AUTHOR

VEDOUCÍ PRÁCE Ing. MARTIN KYSELÁK
SUPERVISOR

BRNO 2007

ZDE VLOŽIT LIST ZADÁNÍ

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Ondřej Matějíček

Bytem: U Nádraží 762, 377 01, Jindřichův Hradec – Jindřichův Hradec II

Narozen/a (datum a místo): 14.4.1984, Jindřichův Hradec

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. Ing. Kamil Vrba, CSc.

(dále jen „nabyvatel“)

Čl. 1
Specifikace školního díla

Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP) - diplomová práce
(dále jen VŠKP nebo dílo)

Název VŠKP:	Řešení hostigových služeb na open source platformách
Vedoucí/ školitel VŠKP:	Ing. Martin Kyselák
Ústav:	Ústav telekomunikací
Datum obhajoby VŠKP:	

VŠKP odevzdal autor nabyvateli v*:

- tištěné formě – počet exemplářů 1
- elektronické formě – počet exemplářů 1

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

* hodící se zaškrtněte

Článek 2
Udělení licenčního oprávnění

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

(z důvodu utajení v něm obsažených informací)

- Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

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

V Brně dne: …….

…….. ……

Nabyvatel Autor

ABSTRAKT

Hlavním cílem této práce je popsat komplexní řešení web-hostingového serveru založeného výhradně na svobodném software. Takový server by měl poskytovat základní služby jako poštovní nebo www server. V práci je popsána instalace hostujícího operačního systému Unixového typu (konkrétně GNU/Linux). Dále jsou v práci rozebrány problémy jednotlivých služeb, vybráni konkrétní zástupci v podobě svobodného software, jejich instalace a konfigurace. I když je v textu popisována realizace konkrétní aplikace, některé části textu obsahují obecné informace , týkající se instalace a zabezpečení linuxových serverů.

KLÍČOVÁ SLOVA

Linux, Hosting, Mail, Www, Gentoo, Server

ABSTRACT

The main point of this work is to describe complex solution of web-hosting server based on free software. This should provide main services such a post or www server. The work describes instalation of Unix(GNU/Linux) operating system. In addition are described individual services, evolved conrete implementations and also is spoken their installation and configuration. Though this text describe implementation of concrete application, some part of thist text contains generally information about installation, configuration and securing of linux servers.

KEYWORDS

Linux, Hosting, Mail, Www, Gentoo, Server

MATĚJÍČEK, O. *Řešení hostingových služeb na open source platformách.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 64 s. Vedoucí diplomové práce Ing. Martin Kyselák.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Řešení hostingových služeb na open-source platformách“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne ....

(podpis autora)

Děkuji vedoucímu Ing. Martinu Kyselákovi za cenné rady a náměty při vedení mé diplomové práce a firmě Maxprogres za poskytnutí hardware a potřebných testovacích prostor.

Seznam kapitol

Seznam obrázků

Seznam tabulek

Úvod

Hostingové servery v dnešní době tvoří základní stavební prvky internetu, co se týče poskytování prostoru pro www stránky, úložný prostor, emailové účty atd. Cílem této diplomové práce je seznámit čtenáře s realizací hostingového serveru, který bude poskytovat základní služby běžně dostupné v prostředí dnešního internetu. Práce je částečně vypracována dle požadavků firmy Maxprogres, pro kterou bude tento server realizován. Pro realizaci projektu byl firmou vyhrazen osobní počítač, který bude umístěn v prostorách jejich web-housingového[1] centra. Firmou byl specifikován požadavek na použití operačního systém Gentoo[2] Linux a veškerý software, který bude na serveru použit , musí být dostupný jako volně šiřitelný.

Systém by měl poskytovat možnost vytváření emailových účtů pro uživatele různých domén. Ke své schránce má mít uživatel přístup přes www rozhraní nebo pomocí emailového klienta a protokolu pop3 nebo IMAP. Protože nevyžádaná pošta a zavirované zprávy tvoří v dnešní době téměř 90% všech emailů a to celosvětově, server by měl poskytovat integrovanou antispamovou a antivirovou kontrolu. Uživatel si dále může zvolit ke svému účtu možnost zřízení prostoru pro své www stránky, kde je dnes již samozřejmostí podpora Php a MySQL databáze. Přístup k vlastním www stránkám je uživateli umožněn pomocí služby FTP[3]. Vytváření nových účtů a správa uživatelů by měla být co nejvíce automatizovaná a realizovaná přes uživatelsky přátelské rozhraní, například www.

Po zvážení všech těchto požadavků bylo potřeba vybrat aplikace, které budou zastupovat jednotlivé služby. Dle požadavků firmy , pro kterou je server realizován, byl jako operační systém vybrán systém Gentoo Linux. Gentoo Linux je distribuce, která je založená na práci se zdrojovými kódy aplikací. Samotné Gentoo je kompletně kompilováno od úplného základu. Obsahuje sice balíčkovací systém, který je velice vyspělý, ovšem balíčky nejsou distribuovány v binární podobě, ale jak již bylo řečeno v podobě zdrojových kódů. Tento systém je tímto předurčen pro zkušenější uživatele, protože veškeré věci si musí administrátor konfigurovat ručně.

Poštovní část serveru tvoří Mail Transfer Agent, který zastupuje program Postfix. O doručování pošty pomocí POP3/IMAP se stará Courier-Imap ve spolupráci s Cyrus-SASL, který slouží pro autentizaci klientů při přístupu uživatelů ke svým emailovým schránkám a posílání zpráv skrz poštovní server. Web-mailové rozhraní pro přístup k emailovým účtům zajišťuje Roundcubemail. Příchozí emaily prochází přes Amavisd-new, kde jsou kontrolovány pomocí antivirového programu Clamav a Spamassassin.

---

1 Prostory určené pro provozování velkého množství serverových strojů. Obvykle disponují velkou konektivitou do sítě internet a poskytují zákazníkům např. klimatizované prostory, záložní zdroje el. energie atd.

2 Svobodný operační systém určený pro nasazení zejména na serverových strojích. Díky tomu, že veškerý software je v této distribuci kompilován pro konkrétní hardware, tato distribuce vyniká především svojí rychlostí a stabilitou.

3 Protokol určený pro přenos souborů po počítačových sítích.

Srdcem druhé části, která poskytuje služby pro tvůrce internetových stránek, je www server Apache. Apache může být používán ve spolupráci s Php a jako databázový server bylo zvoleno MySQL. Databázi je možné ovládat pomocí www rozhraní phpMyAdmin. Přístup uživatele ke svým datům je zajištěn FTP serverem Proftpd.

Protože obsluha systému má být co možná nejvíce automatizovaná a zároveň se předpokládá, že na serveru bude velký počet uživatelů, je výhodné použít systém s virtuálními uživateli[4]. Virtuální systém musí obsluhovat větší počet domén, pod kterými mohou být registrovány stovky uživatelů. Prakticky to znamená, že veškerá autentizace uživatelů vůči jednotlivým službám se děje vůči databázi MySQL, což usnadňuje správu účtu, která by byla jinak velmi obtížná. Zároveň je třeba věnovat zvýšenou pozornost bezpečnost takového systému, což je však základní předpoklad v každém případě.

I když je tento text koncipován jako návod pro realizaci konkrétní aplikace, obsahuje zároveň některé obecné informace, které mohou využít začínající správci serverů postavených na operačním systému Linux. V textu je také ukázáno, jak lze ze vhodně zvoleného bezplatného software vytvořit komplexní fungující server, který je schopný plnit všechny funkce jako servery postavené na placeném software.

Použitá terminologie

Protože tento text popisuje převážně instalaci operačního systému a služeb, je nezbytné , aby pro objasnění vysvětlované problematiky obsahoval ukázky příkazů zadávaných do příkazové řádky. Takový text je umístěn v rámečkách a uvozen znakem „**#“** . Označení příkazu, který je rozdělen na více řádkách , je provedeno znakem „\“ na konci každého řádku.

Dalším typem zvláštního textu jsou komentované výpisy důležitých konfiguračních souborů. Tyto výpisy jsou umístěny ve stejném rámečku jako příkazy, ale neobsahují uvozovací znak „#“. Pro označení komentářů jsou použity znaky „//“ pokud se jedná o jednořádkový komentář, a dvojice znaků /* .... */ pro označení víceřádkového komentáře. Použité konvence by měly být v textu jasné a intuitivní.

4 Virtuální uživatelé nemají přímý přístup Linuxového systému přes konzoli či ssh-klienta. Přístup k systému je umožněn pouze pro konkrétní službu, kterou na serveru využívají. Správa virtuálních uživatelů je snažší a je vyloučena možnost lokálních útoků na systém.

1. Instalace a konfigurace systému Gentoo

1.1 Úvod do instalace systému

Popisování podrobné instalace operačního Gentoo Linux je nad rámec této diplomové práce (více podrobností [7,8]) a budou zde uvedeny pouze základní informace pro zprovoznění systému, rozdělení jeho disků, vytvoření diskových oddílů, instalace potřebných systémových balíčků a základní zabezpečení systému pro potřebu realizace hostingového serveru.

Pro instalaci operačního systému Gentoo Linux je potřeba nejdříve získat zdrojové kódy, které jsou k volně k dostání na síti internet a distribuovány pomocí serverových zrcadel[5]. Distribuci je možné získat v různých podobách v závislosti na tom, zda má uživatel při instalaci přístup k síti internet a podle toho pro jaký účel chce systém využívat. Instalace je rozdělená na dvě části, kdy v první je spuštěn pouze základní systém Gentoo z instalačního LiveCd[6].

V prvním kroku je dokončena instalace základního systému a je zkompilováno jádro pro konkrétní hardware, který je používán. V této fázi už zbývá pouze nakonfigurovat zavaděč systému a nainstalovat ho do MBR[7] pevného disku.

Po zavedení plnohodnotného operačního systému je potřeba nainstalovat základní systémové nástroje, které pomůžou zabezpečit systém, jako například systémový logger[8] pro zaznamenávání hlášení systému a jednotlivých programů. Důležitá je také instalace a správná konfigurace firewallu[9].

Po dokončení všech těchto kroků je systém připraven pro instalaci jednotlivých služeb, jak bude popsáno v dalších kapitolách.

1.2 Rozdělení pevného disku

Důležitou věcí , kterou je potřeba při instalaci operačního systému zvážit, je rozdělení pevného disku. Veškeré operace s diskovými oddíly se při instalaci Gentoo provádějí ručně pomocí programu Fdisk, který je dostupný v systému spuštěném z instalačního LiveCd. Konkrétní rozdělení oddílů na disku je vždy individuální záležitost, protože např. systém, který bude používán jako směrovač, má zcela odlišné požadavky na diskový prostor než poštovní a web-hostingový server. Systém pro svoji funkci potřebuje pouze 3 základní oddíly.

5 Seskupení serverů poskytujících určitá stejná data nebo software, rozprostřená po celém světě z důvodu rozprostření zátěže při masovém stahování dat.

6 Distribuce Linuxu určená k nabootování systému z CD nebo DVD mechaniky. Tyto distribuce jsou určeny pro uživatele, kteří se chtějí seznámit s konkrétní distribucí, nebo se dají využít pro záchranu dat při havárii. Obecně jsou veškerá provedená nastavení v takovém systému po restartování systému zapomenuta.

7 Master Boot Record, je první stopa pevného disku, která většinou obsahuje systémový zavaděč a tabulku rozdělení pevného disku

8 Systémová služba, která sbírá data od jednotlivých programů a podle typu a nastavení je poskytuje administrátorovi

9 Software, který filtruje síťový provoz na jednotlivých portech systému podle pravidel zadaných administrátorem systému.

Základními oddíly jsou:

- Oddíl pro kořenový adresář, tzv. root
- Bootovací[10] oddíl pro zavádění systému
- Oddíl použitý jako odkládací, takzvaný Swap[11]

Systém by samozřejmě při této konfiguraci fungoval, ale za předpokladu, že na tento systém bude přistupovat mnoho uživatelů, je vhodné pro některé adresáře vytvořit oddělený oddíl. Problémy by totiž nastaly, pokud by došlo k zaplnění pevného disku. Takovému stavu by samozřejmě mělo být předcházeno opatřeními provedenými v příslušných částech obsluhujícího software, ale vždy je potřeba počítat s příchodem neošetřené události. Pokud by například nějaká chyba vygenerovala velké množství služebních hlášení do systémového loggeru, začal by neúměrně narůstat logovací soubor, kde jsou tyto zprávy uloženy. V případě, že by oddíl pro data uživatelů a jejich poštu nebyl oddělený od částí souborového systému, se kterým pracuje systém, nebylo by možné pro uživatele garantovat dostatek prostoru pro jejich data, což je v komerčním prostředí nepřípustné. Když ale vytvoříme pro tyto adresáře vlastní oddíl, k této situaci nemůže dojít. Pro každý oddíl také můžeme vybrat individuální souborový systém, který nejlépe odpovídá účelu jeho používání. Můžeme zvýšit i bezpečnost systému tím, že některé oddíly připojíme s parametry, které omezí množinu povolených operací s těmito systémy.

Firma Maxprogres dala k dispozici pevný disk o velikost 40 GB, který jsem po všech uváženích rozdělil pomocí programu **fdisk** následujícím způsobem

Oddíl	Systém souborů	Velikost	Popis
/dev/hda1*	Ext2	64MB	Bootovací oddíl pro zavedení systému
/dev/hda2	Swap	512MB	Odkládací oddíl Swap
/dev/hda3	Ext3	8GB	Oddíl pro kořenový adresář systému /
/dev/hda4	-	-	Extended oddíl pro další oddíly
/dev/hda5	Ext3	5GB	Adresář /tmp
/dev/hda6		5GB	Adresář /var
/dev/hda7	Ext3	20GB	Adresář /home pro uživatelská data

Tab. 1.1 Rozdělení pevného disku

Po rozdělení disku je potřeba nastavit oddíl /dev/hda1 jako bootovací a podle tab. 2.1 vytvoříme na jednotlivých oddílech souborové systémy.

Pro vytvoření systémového souboru typu Ext2 podle [5] použijeme příkaz **mke2fs**, pro vytvoření souborového systému Ext3 použijeme stejný parametr s dodatečnými parametry **mke2fs -j -O dir_index** . Odkládací oddíl pro swap vytvoříme příkazem **mkswap** a aktivujeme ho příkazem **swapon**.

10 Oddíl na kterém je uložen obraz jádra operačního systému. Používá se většinou pouze při zavádění operačního systému

11 Zvláštní systémový oddíl určený jako odkládací oddíl při nedostatku operační paměti systému

Pozn. : Podpora pro souborový systém Ext2 by měla být zakompilována přímo do jádra operačního systému, protože bootovací oddíl je tohoto typu.

Příklad :

```
//vyvtoření souboru systémů na boot oddílu
# mke2fs /dev/hda1

//vytvoření odkládacího   oddílu
#mkswap /dev/hda2

//aktivace odkládacího oddílu
# swapon /dev/hda2
/dev/hda3

//vytvoření ostatních souborových systémů
# mke2fs -j -O dir_index
#   .
#   .
# mke2fs -j -O dir_index /dev/hda7
```

Když jsou jednotlivé souborové systémy vytvořeny, je potřeba je připojit k systému pomocí příkazu **mount**. K tomuto účelu je ve virtuálním systému Gentoo určen adresář /mnt/gentoo(může být samozřejmě použit jakýkoliv jiný adresář). Jako první se připojuje kořenový oddíl umístěný na /dev/hda2 jak je uvedeno v 1.1. Do tohoto adresáře jsou pak připojeny jednotlivé diskové oddíly (/var, /home atd). Samozřejmě je nutné vytvořit potřebnou adresářovou strukturu.

```
//připojení kořenového oddílu
# mount /dev/hda3 /mnt/gentoo

//připojení bootovacího oddílu
# mkdir /mnt/gentoo/boot
# mount /dev/hda1 /mnt/gentoo/boot/

//připojení ostatních oddílů
#   .
#   .
# mkdir /mnt/gentoo/home
# mount /dev/hda7 /mnt/gentoo/home
```

Nyní jsou všechny oddíly připojeny v adresáři /mnt/gentoo. Tím jsme také získali dostatek diskového prostoru pro uložení zdrojových kódů operačního systému.

1.3 Stažení zdrojových kódů OS Gentoo

Jak je uvedeno ve [5], když je rozdělení disku hotové, je možné začít s instalací vlastního operačního systému, která proběhne v několika krocích. Pomocí nástroje **Links**, který poskytuje přímo instalační LiveCd, získáme z jednoho ze zdrojových serverů Gentoo instalační archiv stage3. Tento archiv obsahuje základní instalaci systému. Archiv je potřeba umístit do kořenového adresáře instalovaného systému, který je nyní připojený do /mnt/gentoo.

Po stažení archivu je dobré ověřit pomocí příkazu **md5sum** resp. **sha1sum** integritu a pravost staženého archivu. K tomu účelu je zapotřebí soubor, který obsahuje název archivu a na jeho konci je připojeno .DIGEST. V tomto souboru jsou uloženy otisky archivu v podobě MD5 a SHA1.

Vlastní testování zdrojových souborů potom provádějí tyto příkazy :

```
//Pro MD5
# md5sum -c stage3-x86-2007.0.tar.bz2.DIGESTS

//Pro SHA1
# sha1sum -c stage3-x86-2007.0.tar.bz2.DIGESTS
```

Tyto příkazy nejdříve vypočítají hash[12] souboru stage3-x86-2007.0.tar.bz2 a poté je porovnají s hodnotou uloženou v souboru stage3-x86-2007.0.tar.bz2.DIGESTS. Pokud je nalezena shoda, je vše v pořádku a archiv můžeme extrahovat do kořenového adresáře pomocí archivačního programu **tar**.

Extrahování archivu :

```
# cd /mnt/gentoo
# tar -xpzf stage3-x86-2007.0.tar.bz2
```

Mimo jiné je nutné při extrakci souboru předat programu parametr -p, který zajistí zachování oprávnění k jednotlivým souborům. Význam jednotlivých dalších parametrů je možné získat v manuálových stránkách(viz. man tar) a jejich význam bude popisován i v tomto textu.

Po tomto kroku jsou v kořenovém adresáři (/dev/hda2) nainstalovány potřebné součásti systému Gentoo. V této chvíli systém obsahuje součásti potřebné ke svému běhu, ale neobsahuje součást, která by umožňovala jeho rozšiřování, efektivní správu a aktualizaci. Distribuce Gentoo pro tyto účely využívá program **Portage**, ten však není základní součástí stage3 archivu. Portage je balíčkovací systém a tvoří nezbytnou součást operačního systému Gentoo. Jeho instalace se provádí obdobným způsobem jako instalace vlastního systému, více o tomto kroku je popsáno v [6].

12 Reprodukovatelná metoda pro převod vstupních dat do malého čísla, které vytváří jejich otisk. Tento otisk se označuje jako fingerprig či hash. Funkce se často používá ke kontrole integrity dat.

1.4 Přechod do nově vytvořeného systému

Po splnění všech předchozích kroků je nové prostředí připravené pro uživatelský vstup. Je třeba si uvědomit, že instalovaný systém zatím neobsahuje jádro a zavaděč systému a proto nový operační systém není schopný samostatného nabootování a obsluhy hardware počítače. Z instalačního LiveCd však můžeme příkazem **chroot** změnit aktuální kořenový adresář, a dokončit mimo jiné instalaci linuxového jádra a zavaděče. Před změnou kořenového adresáře je potřeba pro nové prostředí vytvořit a připojit souborový systém /proc, který zpřístupňuje informace poskytované jádrem systému a dále systém /dev který obsahuje zařízení dostupná v systému. Podle [5] se tyto souborové systémy připojují s následujícími parametry:

```
# mount -t proc none /mnt/gentoo/proc
# mount -O bind /dev /mnt/gentoo/dev
```

Pro přechod z prostředí instalačního LiveCd Gentoo do nově nainstalovaného prostředí použijeme příkaz **chroot** :

```
# chroot /mnt/gentoo /bin/bash
```

Po zadání příkazu chroot se systém přepne do nově vytvořeného prostředí, ve kterém se všechny změny ukládají na připojené diskové oddíly a při případném restartování počítače se projeví. Tento systém však ještě není schopen samostatného nabootování, protože neobsahuje jádro operačního systému a proto je potřeba se do tohoto systému při případném restartování opět chrootovat z LiveCd Gentoo.

1.5 Konfigurace Linuxového jádra

Jak již bylo uvedeno, stávající systém zatím neobsahuje jádro, které je srdcem každého linuxového systému. První linuxové jádro vzniklo v roce 1991 a jeho autorem je Linus Torvalds a nyní na jeho vývoji pracují tisíce programátorů z celého světa. Čisté linuxové jádro je označováno jako vanilla-kernel, avšak jeho použití se doporučuje pouze pro speciální účely. Každá z komunit, které vydávají různé distribuce Linuxu, dává k dispozici vlastní aktualizované jádro a popř. do něj přidává další věci, které mají zvýšit jeho funkčnost pro dané aplikace, bezpečnost a výkon. Zdrojové kódy jádra distribuce Gentoo je možné získat pomocí výše uvedeného balíčkovacího systému Portage. Pro práci s Portage slouží příkaz **emerge** jak je popsáno ve [5] .

```
# emerge gentoo-sources
```

Po provedení příkazu se v adresáři /usr/src/linux nachází odkaz na nejnovější verzi jádra, která byla získána pomocí Portage. Před samotnou kompilací jádra je potřeba nastavit jeho

parametry. Pro generování parametrů pro všeobecně použitelné jádro, které bude podporovat nejrůznější typy hardware, obsahuje distribuce pomocný program **genkernel**. Na serveru je však zapotřebí konfigurace ušitá na míru použitému hardware. Tím je docíleno toho, že jádro obsahuje pouze ovladače zařízení, které jsou opravdu potřeba. Takto optimalizované jádro je poměrně malé a poskytuje nejvyšší výkon.

Nezbývá tedy než jádro nakonfigurovat ručně. Při výběru všech potřebných ovladačů je nejdříve potřeba zjistit vše o konkrétním hardware stroje, na který je operační systém instalován. Jak popisuje [10], pokud je totiž nějaký ovladač zapomenut, konkrétní zařízení nebudou pracovat a v horším případě nebude systém vůbec možné nastartovat. Téměř veškerý hardware, který daný počítač obsahuje, lze zjistit např. příkazem **lspci.**

Konfiguraci jádra je možné provést v grafickém rozhraní pomocí příkazu **make menuconfig.** Menuconfig umožňuje nastavení linuxového jádra v prostředí ncurses[13].

```
# cd /usr/src/linux
# make menuconfig
```

Po vybrání všech potřebných ovladačů je aktuální konfigurace uložena do souboru a a jádro je možné i s ovladači zkompilovat. Jak je uvedeno v [10] , po kompilaci musí být ovladače – jaderné moduly pro konkrétní jádro nainstalovány[14].

```
# make && make modules_install
```

Komprimovaný obraz zkompilovaného linuxového jádra operačního systému se nyní nachází v adresáři /usr/src/linux/arch/i386/boot/bzImage, odkud je potřeba ho nakopírovat do bootovacího oddílu, kde ho hledá zavaděč operačního systému. Ten jej extrahuje a zavede do operační paměti počítače. Při kopírování jádra je dobré ho pojmenovat např. podle jeho aktuální verze. To je důležité např. při aktualizaci na novější verzi jádra.

```
# cp /usr/src/linux/arch/i386/boot/bzImage
/boot/kernel-2.6.23-r9
```

Tímto je jádro připravené na zavedení a je potřeba nakonfigurovat zavaděč operačního systému, aby jádro správně zavedl.

13 Knihovna poskytující rozhraní pro tvorbu aplikací v textovém režimu běžících v Unixovém terminálu. Tato knihovna je součástí projektu GNU.

14 Jádro operačního systému můžeme vytvoří jako monolitické, modulární a nebo hybridní. V monolitickém jádru jsou všechny potřebné ovladače zakompilovány přímo v jádru. V modulárním jádru nejsou ovladače přímo součástí jádra, ale jsou do něj zaváděny až v případě potřeby jako tzv. **moduly**. Hybridní jádra jsou kombinací předchozích typů.

1.6 Konfigurace zavaděče operačního systému

1.6.1 Instalace zavaděče

Zavaděč operačního systému je program, který je umístěný většinou v úvodních sektorech pevného disku (typicky MBR), jak popisuje [17]. Zavaděč musí být umístěn na oddílu, který je označen jako bootovací, v našem případě to je oddíl /dev/hda2. Řízení zavaděči předává BIOS po ukončení úvodní inicializace hardwaru.

Zavaděč má za úkol zavést Linuxové jádro do operační paměti a poté mu předat řízení. V Linuxu se používají dva základní zavaděče . Starší z nich je program LILO, který už je v dnešní době považován za zastaralý, novější a více propracovaný program je GRUB, který používá většina moderních Linuxových distribucí. GRUB oproti LILO nevyžaduje při změně konfigurace opětovnou instalaci do MBR a má vestavěný příkazový interpret, díky kterému je možnost v případě havárie nastalou situaci lépe řešit.

V OS Gentoo je podle [5] doporučeno použití zavaděče GRUB, který do systému nainstalujeme pomocí Portage.

```
# emerge grub
```

GRUB pro svoji práci potřebuje konfigurační soubor grub.conf, který musí být umístěn v adresáři /boot/grub/grub.conf. V grub.conf může být uvedeno mnoho konfiguračních parametrů, více o nich je možno najít v [19], v tomto textu budou vysvětleny pouze parametry nezbytné pro správné zavedení systému.

GRUB používá rozdílné označení diskových oddílů oproti značení běžně používanému v Linuxu. Zavaděči GRUB je potřeba předat informaci o tom, kde má hledat jádro operačního systému, respektive disk a oddíl. Pevné disky, ať už IDE, SCSI nebo SATA jsou v terminologii GRUB značeny jako „hd“, jako další následuje pořadí zařízení na sběrnici. To je závislé na tom, k jakému kanálu je zařízení připojené a na aktuální konfiguraci BIOSu. Poslední číslo udává číslo diskového oddílu. Obě čísla se počítají od 0. Disk označený v Linuxu jako hda1 je pro GRUB viditelný jako zařízení hd0,0. Tuto informaci předáme zavaděči GRUB pomocí parametru **root.** Dále je potřeba jádru předat celou cestu k obrazu jádra operačního systému pomocí parametru **kernel**. Zároveň můžeme jádru předat další parametry, obecně je třeba předat alespoň cestu ke kořenovému oddílu, který je umístěný na /dev/hda3.

```
//výpis z grub.conf
    .
root (hd0,0)
kernel /boot/kernel-2.6.23-r9 root=/dev/hda3
//konec výpisu
```

Podle [19], takto nakonfigurovaný GRUB můžeme nainstalovat do MBR pevného disku pomocí příkazu **grub-install.** Ten potřebuje informace o připojených souborových systémech, které jsou v případě zavedeného systému dostupné v souboru /etc/mtab. V našem případě, kdy jsme do nového systému vstoupili pomocí příkazu chroot, jsou informace o připojených souborových systémech dostupné v /proc/mounts a proto je odsud pouze vykopírujeme.

```
# cat /proc/mounts /etc/mtab
# grub-install
```

Tímto krokem je zavaděč nainstalován v zaváděcím sektoru pevného disku a po restartování počítače by byl zaveden nově instalovaný operační systém.

1.6.2 Záloha systémové zavaděče

Je potřeba si uvědomit, že zavaděč systému je zvláště citlivé místo a při jeho poškození nebude možné systém vůbec zavést. Z toho důvodu je výhodné si zaváděcí sektor zálohovat pro případ havárie. K tomu můžeme využít např. program **dd,** který provádí nízko-úrovňové kopírování dat z pevného disku. Předtím než si MBR zazálohujeme, je nutné znát jeho strukturu.

MBR má velikost 512bytů a obsahuje 3 části. V první části, která je velká 446bytů, je uložená primární část bootovacího programu. Další část o velikost 64bytů obsahuje tabulku oddílů, kterou je také výhodné zálohovat. Poslední 2 byty v MBR obsahují informaci o přítomnosti zavaděče a jeho integritě. Více o příkazu dd nalezneme pomocí man dd.

Adresa		Význam	Velikost v bajtech
Hex	Dec		
0000	0	Kód zavaděče	446
01EE	446	Tabulka oddílů	64
01FE	510	Podpis zavaděče	2

Tab 1.2 Struktura Master Boot Record

```
# dd if=/dev/hda of=/mbr.zaloha bs=512 count=1
```

Příkaz provede zálohu zavaděče systému s tabulkou oddílů do souboru mbr.záloha do kořenového oddílu. Pro takto vytvořený soubor potom vytvoříme kontrolní otisk pro větší jistotu.

```
# sha1sum /mbr.zaloha > /mbr.zaloha.DIGEST
```

Tyto dva soubory uložíme na bezpečné místo. V případě potřeby nejdříve ověříme integritu zálohy a zavaděč obnovíme následujícím příkazem :

```
# sha1sum -c /mbr.zaloha.DIGEST
# dd if=/mbr.zaloha of=/dev/hda bs=512 count=1
```

1.6.3 Zabezpečení zavaděče

Linuxový systém je možné při startu systému spustit v tzv. **Single-user** režimu. V tomto režimu může se systémem pracovat pouze superuživatel. Síťové služby nejsou aktivní a heslo administrátora nemusí být vyžadováno. To je potenciální riziko, pokud by útočník získal fyzický přístup ke stroji. Do single-user režimu vstupujeme pomocí zavaděče systému, kde předáme jádru parametr single a systém nabootujeme. Z tohoto důvodu zavaděč systému GRUB zabezpečíme proti zásahům útočníka pomocí hesla. Heslo je uloženo jako MD5 hash. Pro získání hashe našeho hesla můžeme použít přímo vestavěný příkazový řádek zavaděče GRUB. Pomocí příkazu md5crypt si heslo vygenerujeme a pomocí parametru –password md5 toto heslo předáme zavaděči operačního systému.

```
# grub //spuštění příkazové řádky GRUB
> md5crypt //vygenerování hesla
```

Takto nastavený zavaděč bude vyžadovat zadání hesla předtím než umožní editaci parametrů předávaných jádru operačního systému.

1.7 Nastavení přípojných bodů

Jak popisuje [11], jsou v Linuxu všechny diskové oddíly, připojené k systému, uvedeny v souboru /etc/fstab[15]. Tento soubor z rozbaleného archivu Gentoo-stage3 obsahuje soubor, který není implicitně možné použít a je třeba zásah uživatele. Jednotlivé řádky zapsané v tomto souboru tvoří jednotlivé svazky připojené k operačnímu systému. Každý řádek obsahuje 6 položek, které mají podle [11] následující význam :

1. Položka označující zařízení, které budeme k systému připojovat, např. běžný oddíl disku
2. Adresář do kterého bude svazek připojen. Pro typ swap musí být zadáno **none.**
3. Typ souborového systému, který připojujeme. Kompletní seznam jádrem podporovaných systémů je možné nalézt v souboru /proc/filesystems. Pokud potřebujeme podporu i pro jiné souborové systémy, než naše jádro podporuje, musíme buď překompilovat jádro s podporou nového ovladače a nebo zkompilovat ovladač jako modul jádra a modul do jádra operačního systému zavést.
4. Volby využívané programem **mount** při připojování oddílů. Parametrů může být více a musí být odděleny čárkou. Každý typ souborového systému má své vlastní parametry. Jejich kompletní seznam můžeme nalézt s pomocí nápovědy **man mount.** Mezi nejzákladnější podle [11] patří :
    - noauto(auto) – svazek není připojen automaticky při startu systému
    - nouser(user) – svazek nemohou připojovat a odpojovat běžní uživatelé
    - isocharset – znaková sada pro názvy souborů
    - noatime(atime) – nedochází k aktualizaci času přístupu k systému souborů, čímž

15 FileSystem table, neboli tabulka souborových systémů. Má za úkol popsat jednotlivé diskové svazky v systémech Unixového typu.

se zrychlí přístup

- nodev(dev) – zakáže interpretování blokových nebo znakových zařízení
- suid – povoluje spouštět soubory, které mají nastavený bit suid
- noexec(exec) – nepovoluje spouštění souborů na tomto souborovém systému
- ro - read only - systém souborů je připojen pouze pro čtení
- rw – read write – systém souborů je připojen pro čtení i zápis
- **defaults** sdružuje některé běžné používané parametry rw, suid, dev, exec, auto, nouser.

5. Parametr používaný programem **dump**[16]**.** Standartně je možné nechat na tomto místě 0, což znamená že tento systém nebude zálohován. Hodnota 1 nařizuje zálohování souborového systému.
6. Poslední z parametrů, který používá program **fsck**[17]**,** určuje v jakém pořadí bude kontrolována konzistentnost jednotlivých souborových systémů. Kořenový oddíl by měl mít nastavenou hodnotu 1, ostatní file-systémy hodnotu 2 nebo 0(žádná kontrola se neprovádí). Mimo pravidel pro všechny diskové oddíly je potřeba připojit i zvláštní systémové oddíly.

Jsou to souborový systém **/proc**, které jak bylo uvedeno dříve využívá ke své činnosti jádro operačního systému, a dále souborový systém **/tmpfs**[18].

Příklad souboru /etc/fstab

```
//bootovací oddíl
/dev/hda1     /boot     ext2      defaults,noatime    0 2

//swapovací oddíl
/dev/hda2     none      swap      sw                  0 0

//kořenový oddíl
/dev/hda3     /         ext3      noatime             0 1

/*oddíl /tmp, v tomto adresáři není potřeba spouštět žádné
soubory ani připojovat žádné další zařízení */
/dev/hda5     /tmp      ext3      noexec,nodev,nosuid 0 0

//oddíl var
/dev/hda6     /var      ext3      default,noatime     0 0
```

16 Program Dump se stará o zálohování souborových systémů

17 Program fsck provádí kontrolu konzistence souborového systému. Obvykle se fsck spouští při startování systému a odhaluje chyby v souborovém systému vzniklé např. nekorektním ukončením systému nebo při výpadku dodávky eletrické energie.

18 Tmpfs je souborový systém používaný pro dočasné soubory. Tento systém se však nenachází na fyzickém disku, ale ve virtuální paměti počítače. V Linuxu je tento systém většinou připojován do adresáře /dev/shm.

```
//oddíl home
/dev/hda7     /home     ext3      default,noatime     0 2

//oddíl proc
proc          /proc     proc      default             0 0
```

1.8 Instalace důležitých systémových programů

Nově instalovaný operační systém neobsahuje nástroje, které jsou pro administrátora v průběhu jeho práce a údržby systému nezbytné. Tyto nástroje administrátorovi pomáhají zvýšit zabezpečení serveru a umožňují monitorovat důležité části celého systému. Nejsou součástí stage3 archivu, protože implementací od jednotlivých nástrojů existuje více a uživatel má při instalaci možnost si konkrétní implementaci vybrat. V následujícím výpisu jsou uvedeny některé z nástrojů doporučené v [5] , plus některé další.

1.8.1 Systémový logger Syslog-ng

Systémový logger je program, který shromažďuje data od jednotlivých služeb, démonů, programů a jádra operačního systému. Tyto informace potom logger zaznamenává do souborů, tzv. logů, které jsou většinou umístěny v adresáři /var/log/. Konkrétní umístění souboru s logy od jednotlivých programů je dáno

- nastavením konkrétního programu, služby
- nastavením konkrétního loggeru
- konkrétní implementací systémového loggeru

V operačním systému Gentoo je na výběr z několika systémových loggerů např. Sysklogd, Syslog-ng, Metalog a další. Ze zmíněných zástupců jsem zvolil logger Syslog-ng. Nástroj nainstalujeme pomocí emerge a přidáme ke službám, které jsou automaticky spouštěny po startu systému. Zároveň se systémovým loggerem je potřeba nainstalovat nástroj, který bude provádět tzv. rotaci systémových logů.

Rotací systémových logů rozumíme proces, při kterém dochází k zálohování současných souborů s logy a vytvoření nových prázdných souborů na jejich místo. Takto rotované logy zajišťují redukování velikosti jednotlivých souborů a zároveň zajistíme, že starší soubory s logy jsou průběžně mazány. Soubory se staršími logovacími údaji se uchovávají kvůli úspoře místa v archivované podobě. Nástroj zajišťující rotaci systémových logů se jmenuje **Logrotate** a je k dispozici ve stromu Portage.

```
#emerge syslog-ng
#emerge logrotate
#/etc/init.d/syslog-ng start
#rc-update add syslog-ng default
```

1.8.2 Cron

Nástroj Cron, jak popisuje [9], slouží pro plánované spouštění příkazů. Umožňuje příkazy spouštět v předdefinovaných adresářích např. každou hodinu, den, měsíc atd. Také je možné definovat pomocí příkazu **crontab** vlastní čas spouštění. Programem Cron je např. možné zajistit pravidelné spouštění nástroje Logrotate. Aby Cron mohl obstarávat pravidelné spouštění programů, musí být na serveru trvale spuštěn - jako služba.

```
#emerge vixie-cron
#/etc/init.d/syslog-ng start
#rc-update add vixie-cron default
```

Přidávání záznamů do programu Cron lze provádět výhradně příkazem crontab -e. Záznamy vložené do Cron mají 6 parametrů, které jak popisuje [9] mají následující význam :

1. Minuta
2. Hodina
3. Den v měsíci
4. Měsíc
5. Den v týdnu
6. Program, který chceme spustit

Každý z parametrů můžeme nahradit znakem *, ten znamená, že parametr nechceme definovat. Jako příklad můžeme použít skript, který provede každý den v noci zálohu databáze uživatelů.

1.8.3 Iptraf

Iptraf je analyzátor síťového provozu. Pomocí tohoto konzolového programu můžeme snadno a přehledně sledovat komunikaci, která probíhá přes síťové rozhraní. Využijeme ho například při ladění firewallu, řešení problému při přetížení linky, zjišťování příčiny generování nadměrného síťového provozu, odhalování DoS[19] útoků a další. V Iptraf lze nastavovat různé filtry, které usnadňují práci a orientaci ve spojeních při velkém provozu.

```
#emerge iptraf
```

1.9 Nastavení firewallu

K základnímu zabezpečení každého serveru patří správně nainstalovaný a nakonfigurovaný firewall[20]. V moderních distribucích Linuxu se k nastavení stavového firewallu používá nástroj Iptables, který je dostupný v jádrech řady 2.6 a vyšších. V Gentoo je Iptables dostupný přes Portage.

19 **DoS**(Denial of Service) – útok odepřením služby

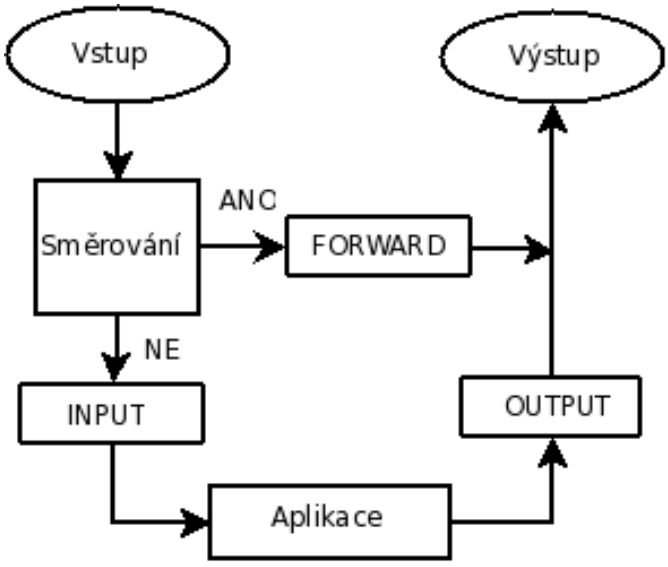
20 Firewall je hardwarové nebo softwarové síťové zařízení, které zabraňuje nevyžádané síťové komunikaci. Firewall může pracovat na různých vrstvách síťového modelu, podle pravidel daných administrátorem systému

```
# emerge iptables
# /etc/init.d/iptables start
```

K nastavování firewallu je možno přistupovat různými způsoby, jak popisuje [4] a [12]. Například zakázat všechny porty, které nejsou využité a ostatní nezakazovat. Obecně je ale lepší nastavit politiku firewallu tak, že všechno co není povoleno ,je zakázáno. Tak by se nemělo stát, že některý port necháme nezabezpečený. Pravidla firewallu je možné zadávat přímo v příkazové řádce. Výhodné je napsat dávkový soubor, ve kterém nejdříve nastavíme restriktivní bezpečnostní politiku firewallu a potom povolujeme jednotlivé porty dle různých pravidel.

Firewall je možné editovat z příkazové řádky, to ale není při větším množství pravidel přehledné a při každém restartování by bylo nutné pravidla firewallu definovat znovu. Proto si vytvoříme jednoduchý skript s pravidly firewallu. Systém se potom nakonfiguruje tak, aby soubor s pravidly firewallu spustil při každém spuštění systému. To administrátorovi usnadňuje práci.

Detailní popis fungování firewallu, nebo jeho konkrétní implementace balíčku Iptables by bylo nad rámec toho textu. Detailní popis i s příklady obsahuje [12] a [13]. Jako základ pro zabezpečení serveru nám postačí fakt, že jakýkoliv provoz přicházející na síťové rozhraní serveru nejdříve prochází systémem řetězců obsahujících pravidla. Tyto řetězce tvoří tzv. filtrovací tabulku firewallu a ovlivňují to, jak bude z řetězcem zacházeno. Obecně tvoří filtrovací tabulku tři řetězce, jak popisuje i [13].



obr. 1.1 Průchod paketu firewallem

- řetězec INPUT – pakety přicházející na síťové rozhraní
- řetězec FORWARD – pakety určené pro směrování
- řetězec OUTPUT – pakety vystupující z rozhraní

V případě hostingového serveru k žádnému směrování nedochází a proto je pro tento případ řetězec FORWARD nepodstatný. Největší význam má řetězec INPUT, který ovlivňuje dostupnost portů, na kterých naslouchají jednotlivé služby, z prostředí sítě internet.

Následuje příklad vytvořený pomocí [4] a [12], který ukazuje možné nastavení firewallu vhodné pro realizovaný hostingový server. Tato pravidla uvažují povolení všech potřebných služeb. Jednotlivé položky v konfiguračním souboru jsou okomentovány.

```
//Odstranění všech stávajících pravidel firewallu
iptables -F
iptables -X

//Nastavení restriktivní politiky firewallu(co není povoleno je
//zakázáno)
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

//Vytvoření nového řetezce pro pakety protokolu TCP
iptables -N tcp_segmenty
iptables -A INPUT -p TCP -i eth0 -j tcp_segmenty

//Povolení služeb Http,Ftp,Smtp,Pop3 a Imap
iptables -A tcp_segmenty -p tcp --dport 80 -j ACCEPT
iptables -A tcp_segmenty -p tcp --dport 21 -j ACCEPT
iptables -A tcp_segmenty -p tcp --dport 25 -j ACCEPT
iptables -A tcp_segmenty -p tcp --dport 110 -j ACCEPT
iptables -A tcp_segmenty -p tcp --dport 143 -j ACCEPT

//Povolení služby SSH pro jednotlivé IP adresy z kterých chceme
//přistupovat k serveru
iptables -A tcp_segmenty -s zdroj.adr. -p tcp --dport 22 -j ACCEPT
                         .
//Povolení služebního protokolu ICMP
iptables -A INPUT -p ICMP  --icmp-type 0 -j ACCEPT
iptables -A INPUT -p ICMP  --icmp-type 3 -j ACCEPT
iptables -A INPUT -p ICMP  --icmp-type 8 -j ACCEPT
iptables -A INPUT -p ICMP  --icmp-type 11 -j ACCEPT

//Povolení všech paketů přicházejících na rozhraní loopback
iptables -I INPUT -p ALL -i lo -j ACCEPT
iptables -A OUTPUT -p ALL -o lo -j ACCEPT

//Zaznamenání všech paketů které přijdou na vstup firewallu
iptables -A INPUT -j LOG

//Povolení všech odchozích spojení
iptables -A OUTPUT -o eth0 -j ACCEPT
//Povolení příchozích spojením, která již byly navázány
iptables -A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j
ACCEPT
//############################blacklist#################################
//Zakázání příchozích spojení s konkrétních IP adres nebo
//sítí,které  #chceme blokovat nebo z nich byly vedeny nějaké útoky
iptables -I INPUT -i eth0 -s source ip  -jDROP
```

Konfiguraci firewallu je možné zkontrolovat pomocí příkazu iptables s parametrem -L.

```
#iptables -L -vn
```

Pozn.: Parametr v zajistí zobrazení rozšířených informací a parametr n zabrání překládání ip adres na doménové jména. Tím zpřehledníme a zrychlíme výpis pravidel firewallu.

Správné nastavení firewallu je klíčovou částí v zabezpečování serveru a proto je vhodné jeho nastavení věnovat dostatečnou pozornost a danou problematiku nastudovat hlouběji. V úvahu také přichází použití pomocných nástrojů, které uživateli usnadňují správu a definování nových pravidel firewallu. Např. programy Firestarter, Shorewall a další.

1.10 Zálohování celého systému

Po dokončení všech úkonů instalace, kdy vše pracuje tak jak má, je výhodné před instalací dalších síťových služeb systém kompletně zazálohovat. Záloha MBR pevného disku se zavaděčem operačního systému a s tabulkou rozdělení pevného disku byla popsána v kapitole 1.1.6. Zálohu všech diskových oddílů je možné provést při běžícím systému. Využijeme k tomu např. program **tar** s příslušnými přepínači, které zajistí zkomprimování výsledného souboru zálohy. Dále je potřeba zachovat příslušná oprávnění k souborům. Při zálohování se nedoporučuje zálohovat některé speciální typy adresářů, např. /proc , /dev, /sys nebo /tmp. Tyto obsahují informace, které není nutné zálohovat, protože je generuje běžící jádro a nebo obsahují dočasné soubory, které při obnově systému nejsou potřeba.

```
# tar -cpvzf --exclude /dev --exclude /tmp --exclude /proc  \
     --exclude /sys --exclude /gentoo.zaloha  gentoo.zaloha /
```

Význam jednotlivých přepínačů :

- c - vytvoření nového archivu
- p - zachování všech informací o přístupových právech souborů
- v - zapne podrobný výpis prováděných operací
- z - soubor bude komprimován pomocí programu gzip
- f - jako vstup bude určen soubor
- exclude - vynechá uvedený adresář

Stejně jako v kap. 1.1.6. je vhodné vytvořit otisk takto zálohovaného systému, tím si při obnově systému můžeme být jisti, že archiv není poškozen či záměrně pozměněn. Archivu je dobré udělat několik kopií a bezpečně uložit.

```
# sha1sum /gentoo.zaloha > /gentoo.zaloha.DIGEST
```

Archiv ověříme následujícím příkazem

```
# sha1sum -c /gentoo.zaloha.DIGEST
```

Obnovu systému na předpřipravený pevný disk provedeme opět pomocí programu tar.

```
# tar -xvpf /gentoo.zaloha /
```

- x - přepínač pro extrahování

1.11 Alternativní možnosti instalace

1.11.1 Použití logických dynamických disků LVM

Jak uvádí [2] LVM je implementace správce logických disků pro Linux. Použití logických disků řeší problém, kdy potřebujeme přerozdělit velikost diskového oddílu. LVM přidává vrstvu mezi fyzické médium a operační systém. V logických jednotkách lze snadno manipulovat s fyzickými disky (přidávat a odebírat je) a tím měnit jejich velikost.

LVM rozeznává tři základní typy svazků

- Fyzický svazek – oddíl na fyzickém disku
- Skupina svazků – sdružuje jednotlivé fyzické svazky do jednoho celku
- Logický svazek – definován uvnitř skupiny, pro oper. systém se jeví jako fyzický disk

1.11.2 Použití diskových polí RAID[21]

V komerčních aplikacích, kdy je potřeba zajistit, že uživatelé nepřijdou o svá data, není možné spoléhat na jednu fyzickou jednotku. Použitím více fyzických jednotek zajistíme větší odolnost vůči ztrátě dat, nebo můžeme zvýšit výkon systému, co se týče diskových operací. Pro tyto účely slouží diskové řadiče RAID, jak popisuje i [18]. Tyto jsou realizovány buď hardwarovým způsobem, takový RAID musí být podporován příslušnou základní deskou serveru, a nebo jako softwarový RAID, kdy příslušné operace zajišťuje jádro operačního systému. Hardwarový RAID zpravidla dosahuje vyšších výkonů než softwarový RAID, to je však vykoupeno vyšší cenou základní desky serveru. Linux ve většině případů používá softwarový RAID, který již v dnešní době může konkurovat hardwarové implementaci systému RAID.

Existuje několik typů systému RAID, každý z nich funguje odlišným způsobem a má různé využití.

Standardní typy jsou :

1. RAID 0 – tento typ umožňuje pracovat v režimu zřetězení nebo prokládání. V tomto režimu

21 **Redundant Array of Independent Disks** (vícenásobné diskové pole nezávislých disků). Typ diskového řadiče zajišťující koordinovanou spolupráci dvou a více diskových jednotek.

nejsou data odolná vůči poruše pevného disku. Výhoda v režimu zřetězení spočívá ve vytvoření jednoho velkého logického disku a v případě prokládání ve zrychlení čtení a zápisu velkých souborů.

2. RAID 1 – v tomto režimu se provádí zrcadlení (mirroring) disků, tzn. že data jsou zaznamenávána na dva disky zároveň a v případě výpadku jednoho z disků jsou k dispozici data na druhém disku. Nevýhodou toho typu je pouze 50% využití kapacity nosičů.
3. RAID 3 – je potřeba N+1 disků a pro jejich zabezpečení je využito vlastností operace XOR. N+1 disk je využit jako paritní disk a při výpadku jednoho z disků je možné data pomocí ostatních disků a tohoto paritního disku data obnovit. Při výpadku paritního disku nedochází ke ztrátě dat vůbec.
4. RAID 5 – je podobný typu RAID 3, ale odstraňuje nutnost použití paritního disku tím, že jsou redundantní informace uloženy střídavě na všech discích.
5. Ostatní nestandardní typy RAID vznikají většinou skládáním předchozích typů. Více o nich je možné najít v [18].

2. Popis jednotlivých služeb

2.1 FTP Server

Služba FTP je základem každého web-hostingového serveru, protože umožňuje uživateli přístup k jeho webové prezentaci, která je umístěna na vzdáleném hostingovém serveru. Uživatel takto může přistupovat ke svému účtu kdekoliv, kde má přístup na internet pomocí FTP klienta. Zároveň je však třeba dbát na to aby tato služba byla dobře chráněna proti případným útokům, v opačném případě by mohlo dojít ke ztrátě nebo zcizení uživatelových dat, kompromitaci jeho webové prezentace atd.

V oblasti otevřeného software je takovýchto FTP serverů hned několik. Mezi ty nejznámější můžeme zařadit **Vsftpd, Pure-FTP , WU-FTP, Proftpd** a další. Všechny z těchto jmenovaných serverů prošly dlouhým vývojem, a jak uvádí [7], některé z nich obsahovaly řadu bezpečnostních chyb, zejména typu buffer-overflow[22], které mohly vést nejen k získání přístupu k uživatelským datům, ale často i k získání příkazové řádky s právy roota. Z těchto zástupců jsem pro server vybral server Proftpd, který je vysoce bezpečný, snadno konfigurovatelný a obsahuje velké množství přídavných modulů použitelných pro různé typy potřeb uživatele či administrátora.

Můžeme předpokládat, že na webhostingovém serveru bude velký počet uživatelů, a proto je potřeba vybrat způsob, jakým se budou autorizovat[23] a autentizovat[24] při vstupu do systému. Standardně může Proftpd uživatele ověřovat jako klasické UNIXové uživatele, což znamená, že ověřování uživatelů se provádí pomocí souborů /etc/passwd respektive /etc/shadow . V dnešní době je tato metoda již zastaralá a nedostatečná a proto má Proftpd možnost využívat tzv. Mechanismus PAM[25](správa autentizačních mechanismů), který podle [7] zapouzdřuje proces autentizace a příslušná aplikace pak pouze obdrží výsledek, zda autentizace uživatele proběhla v pořádku.

I když tato metoda je co se týče konfigurace služby nejjednodušší a nevyžaduje téměř žádné úpravy konfiguračních souborů, je pro použití při velkém počtu uživatelů nevhodná. Práce s těmito účty se musí provádět na systémové úrovni a to je pro používání ve spolupráci s webovými aplikacemi příliš složité. Také není vhodné, aby uživatelé měli přímý přístup do systému, tedy možnost přihlašování přes konzoli. Pro tyto případy Proftpd umožňuje využít autentizaci oproti jinému zdroji dat, například vůči databázi jak uvádí [1][15] . Tímto vlastně vytvoříme virtuální uživatele. Pomocí modulu mod_sql_mysql.c jsme schopni zajistit, aby Proftpd prováděl ověření uživatelů vůči databázi MySQL, kde vytvoříme tabulku s příslušnými údaji.

22 **Buffer overflow** je jedna z nejznámějších forem bezpečnostní chyby. Na základě neošetřeného uživatelského vstupu jsou přepsána data v zásobníku a je předána kontrola cizímu kódu.

23 Typicky získání uživatelského jména a hesla od uživatele a následné ověření toho, že má uživatel povolen přístup do příslušné části systému.

24 Ověření, zda má uživatel ,jehož identita již byla ověřena, přístup do určité části systému

25 **PAM**(Pluggable Authentication Modules) - správa autentizačních mechanismů

Dalším užitečným modulem je mod_quotatab_sql.c, který nám umožňuje vyhradit pro uživatele určitou velikost diskového prostoru, který má dostupný pro svojí webovou prezentaci. Informace o dostupném diskovém prostoru uživateli zprostředkovává jeho FTP klient.

2.1.1 Instalace FTP serveru Proftpd v operačním systému Gentoo

Pro instalaci balíčku Proftpd do systému využijeme nástroj **emerge**, který stáhne ze zdrojového serveru příslušný balíček, postará se o veškeré závislosti a podle nastavené proměnné USE[26] pro konkrétní balíček v souboru /etc/portage/package.use provede kompilaci zdrojových kódů programu. Aby byl server zkompilován s příslušnými parametry, musíme nastavit v souboru package.use následující flagy[27]

5. MySQL – zajistí zkompilování modulu mod_sql_mysql.c a následnou možnost autentizace uživatelů oproti MySQL
6. Softquota - zajistí zkompilování modulu mod_quotatab_sql.c a umožní nastavovat uživatelům diskové kvóty
7. -ipv6 – vypne podporu serveru pro ipv6

Poté pomocí příkazu

```
# emerge proftpd
```

nainstalujeme FTP server do systému. Jak je uvedeno v [15], Proftpd hledá svoje konfigurační soubory v /etc/proftpd, konkrétně hlavní konfigurační soubor proftpd.conf. Ten je nutné nejprve vytvořit.

```
# touch /etc/proftpd/proftpd.conf
```

V hlavním konfiguračním souboru potom nastavíme základní parametry, které jsou potřebné k běhu FTP serveru, nastavíme autentizaci uživatelů v MySQL a funkci kvót. Následuje výpis hlavního konfiguračního souboru s okomentovanými parametry.

2.1.2 Základní nastavení

```
// Jméno,identifikace serveru a mail správce
ServerName                      "Maxprogres99 ProFTPD server"
ServerIdent                     on "FTP Server ready."
ServerAdmin                     admin@maxprogres99.cz

// Typ serveru(standalone nebo inetd)
ServerType                      standalone
DefaultServer                   on
```

26 Tyto proměnné jsou klíčová slova obsahující podporu a informace o závislosti pro konkrétní oblast. Pomocí **USE** proměnných předáváme Portage informaci, že chceme v programu podporu pro danou funkci.
27 Flagy nastavují proměnné USE

```
AccessGrantMsg                  "User %u logged in."
DeferWelcome                    off
// Vytvořit domovský adresář uživatele pokud neexistuje
CreateHome                      on

// Uzamknutí uživatele v domovském adresáři a podadresářích
DefaultRoot                     ~ !adm

// Vypnutí DNS dotazů
IdentLookups                    off
UseReverseDNS                   off

// Port na kterém proftpd naslouchá příchozí spojení
Port                            21

// Maska použitá při vytváření nových adresářů
Umask                           022

// Zakázání zobrazování skrytých souborů
ListOptions                     "-a"

// Zakázání přihlášení superuživatele root
RootLogin                       off

// Maximální počet procesů, které může proftpd vytvořit
MaxInstances                    20

// Uživatel a skupina pod kterým je proftpd spuštěn
User                            nobody
Group                           nobody

// Nastavení obecných pravidel pro adresáře. Např povolení
// přepisování souborů atd.
<Global>
  AllowOverwrite                yes
  <Limit ALL SITE_CHMOD>
    AllowAll
  </Limit>
</Global>

// Nastavení Logovacího formátu pro Syslog-ng
LogFormat            default "%h %l %u %t \"%r\" %s %b"
LogFormat            all     "%v [%P] %h %t \"%r\" %s"
```

Pozn. : Tato konfigurace byla převzata z manuálových stránek programu a upravena pro potřeby tohoto serveru. Význam jednotlivých položek je možné najít v [1][15].

2.1.3 Nastavení Proftpd pro MySQL autentizaci a přidělení diskových kvót

Nejprve musíme v MySQL vytvořit tabulky obsahující informace o uživatelích a jim přidělených kvótách. Dále je vhodné v MySQL vytvořit uživatele Proftpd, který bude mít přístup

pouze k těmto tabulkám. Tabulku se seznamem uživatelů pojmenujeme **hostuser**. Tabulka obsahující nastavení limitů pro uživatele je **ftpquotalimits** a tabulka obsahující záznamy o volném místu uživatelů a o přenesených datech se nazývá **ftpquotatallies**(tabulka záznamů). Do konfiguračního souboru serveru Proftpd potom předáme informace o uživateli pro databázi MySQL. Více o významu jednotlivých položek a přístupu k nim je možné najít v [1] a v nadcházejících kapitolách.

Struktury jednotlivých MySQL pro Proftpd

Sloupec	Typ	Nulový	Výchozí	Komentáře
Id_user	int(11)	Ano	-	Id uživatele
Login	varchar(30)	Ano	-	Přihlašovací jméno
Passwd	varchar(80)	Ano	-	Šifrované heslo
Uid	int(11)	Ano	2000	Uid uživatele
Gid	int(11)	Ano	2000	Gid uživatele
homedir	varchar(255)	Ano	2000	Domovský adresář
Shell	varchar(255)	Ano	/bin/false	Cesta k shellu

Tab. 2.1 Struktura tabulky hostuser

Sloupec	Typ	Nulový	Výchozí	Komentáře
name	varchar(30)	Ano	-	Login uživatele
quota_type	enum('user', 'group', 'class', 'all')	Ano	user	Typ kvóty
per_session	enum('false', 'true')	Ano	false	-
limit_type	enum('soft', 'hard')	Ano	soft	Typ limitace
bytes_in_avail	int(10)	Ano	0	Limit uploadu
bytes_out_avail	int(10)	Ano	0	Limit downloadu
bytes_xfer_avail	int(10)	Ano	0	Limit celkově
files_in_avail	int(10)	Ano	0	Limit souborů up.
files_out_avail	int(10)	Ano	0	Limit souborů down.
files_xfer_avail	int(10)	Ano	0	Limit souborů celkově

Tab. 2.2 Struktura tabulky ftpquotalimits

Sloupec	Typ	Nulový	Výchozí	Komentáře
name	varchar(30)	Ano	-	Login uživatele
quota_type	enum('user', 'group', 'class', 'all')	Ano	user	Typ kvóty
bytes_in_used	int(10)	Ano	0	Nahráno na server [bajty]
bytes_out_used	int(10)	Ano	0	Stáhnuto ze serveru [bajty]
bytes_xfer_used	int(10)	Ano	0	Limit celkově [bajty]
files_in_used	int(10)	Ano	0	Nahráno na server [soubory]
files_out_used	int(10)	Ano	0	Stáhnuto ze serveru [soubory]
files_xfer_used	int(10)	Ano	0	Limit celkově [soubory]

Tab. 2.3 Struktura tabulky Struktura tabulky ftpquotatallies

2.1.4 Vytvoření uživatele proftpd v MySQL a přidělení potřebných práv

```
# mysql –u root –p mysql
mysql> GRANT SELECT,INSERT,DELETE,UPDATE ON hosting.hostuser\
       TO proftpd@localhost identified by ,$password‘;

mysql>GRANT SELECT,INSERT,DELETE,UPDATE ON \
      hosting.ftpquotalimits\
      TO proftpd@localhost;

mysql>GRANT SELECT,INSERT,DELETE,UPDATE ON  \
      hosting.ftpquotatallies \
      TO proftpd@localhost;

mysql> flush privileges;
```

Následují konfigurační direktivy přidané do hlavního konfiguračního souboru pro aktivování podpory autentizace uživatelů a kvót.

```
/*Načtení potřebných modulů potřebných  pro MySQL autentizaci
a nastavení diskových kvót */

<IfModule mod_dso.c>
   LoadModule mod_sql.c
   LoadModule mod_sql_mysql.c
   LoadModule mod_quotatab.c
   LoadModule mod_quotatab_sql.c
</IfModule>
```

Hesla uživatelů je možno v databázi ukládat buďto v podobě PlainTextu[28], nebo šifrovaná pomocí algoritmu DES[29]. Seznam všech typů, které je možné při autentizaci použít, je dostupný v [15]. Typ autentizace nastavují direktivy SQLAuthTypes.

```
SQLAuthTypes                    Crypt
SQLAuthenticate                 users*
```

Nastavení připojení k databázi, nastavení uživatele a hesla. Protože heslo je v souboru uloženo v nešifrované podobě, je potřeba, aby vlastníkem souboru proftpd.conf byl superuživatel a oprávnění číst z něj měl pouze on.

```
# chown root:root proftpd.conf
# chmod 600 proftpd.conf
```

```
SQLConnectInfo    hosting@localhost proftpd 286v:sCAaZhbsKZe
SQLDefaultGID     2000
```

28 PlainText – hesla jsou uložena v nezašifrované podobě
29 **DES**(Data Encryption Standard) je symetrická šifra vyvinutá v 70. letech.

```
SQLDefaultUID     2000

/* Nastavení minimálního UID a GID pro které bude uživatel z
databáze mít povolený přístup */
SQLMinUserGID                   2000
SQLMinUserUID                   2000

/* Informace pro proftpd, v jakých sloupcích tabulky má
hledat požadované údaje */
SQLUserInfo   hostuser login passwd uid gid homedir shell

/* Protože systém používá virtuální uživatele, ftp server
nebude vyžadovat aby měl uživatel platný shell. */
RequireValidShell                off[30]
```

2.1.5 Modul mod_quotatab_sql a mod_quotatab

Modul mod_quotatab_sql, který se stará o získávání dat obsahujících informace o limitech uživatele z databáze MySQL, předává tyto data modulu mod_quotatab, který je potom skutečně aplikuje, jak uvádí [1]. Tyto moduly potřebují pro svoji činnost definovat pět řídících direktiv. První z nich je direktiva modulu mod_sql - SQLConnectInfo, která již byla definována v kapitole 2.1.4 a slouží k nastavení připojení modulu k MySQL. Další čtyři direktivy jsou typu mod_sql - SQLNamedQueries a slouží modulu mod_qutatab pro zjištění, v kterých tabulkách jsou uloženy limity pro uživatele a SQL příkazy pro práci s nimi. Data využívají direktivy QuotaLimitTable

První direktiva SQLNamedQueries slouží ke zjištění limitů uživatele z tabulky ftpquotalimits. Dotaz na databázi musí podle [1] vrátit hodnoty v následujícím pořadí :

- přihlašovací jméno
- typ kvóty
- kvóta platná jen pro sezení
- typ limitu
- upload limit v bytech
- download limit v bytech
- přenos celkově v bytech
- limit počtu uploadovaných souborů
- limit počtu stažených souborů
- limit počtu souborů celkově

což odpovídá uspořádání tabulky 2.2 . Konkrétní SQL dotaz

```
//Konstrukce SQL dotazu pro získání limitů uživatele
SQLNamedQuery get-quota-limit SELECT
„* FROM  ftpquotalimits
   WHERE name = ‚%{0}‘ AND quota_type = ‚%{1}‘“

//Definice první direktivy
QuotaLimitTable sql:/get-quota-limit
```

30 Uživatel by musel mít vytvořený systémový účet a možnost přihlásit se do systému

Druhá direktiva slouží pro získání údajů z tabulky ftpquotatallies, kde jsou uloženy záznamy o aktuálním stavu účtu uživatele. Dotaz na databázi musí podle [1] vrátit hodnoty v následujícím pořadí :

- přihlašovací jméno
- typ kvóty
- upload limit v bytech
- download limit v bytech
- přenos celkově v bytech
- limit počtu uploadovaných souborů
- limit počtu stažených souborů
- limit počtu souborů celkově

což odpovídá uspořádání tabulky 2.2 . Konkrétní SQL dotaz

```
//Konstrukce SQL dotazu pro aktuálního stavu konta uživatele
SQLNamedQuery get-quota-tally SELECT
„* FROM  ftpquotatallies
   WHERE name = ‚%{0}‘ AND quota_type = ‚%{1}‘“

//Definice druhé direktivy
QuotaTallyTable sql:/get-quota-tally
```

Třetí direktiva aktualizuje údaje o aktuálním stavu obsazení na účty klienta v tabulce ftpquotatallies.

```
//Konstrukce SQL dotazu pro aktualizaci stavu konta uživatele
SQLNamedQuery update-quota-tally
UPDATE   „bytes_in_used = bytes_in_used + %{0},
          bytes_out_used = bytes_out_used + %{1},
          bytes_xfer_used = bytes_xfer_used + %{2},
          bytes_in_used = bytes _in_used + %{3},
          bytes_out_used = bytes_out_used + %{4},
          bytes_xfer_used = bytes_xfer_used + %{5}
WHERE     name = ‚%{6}‘ AND quota_type = ‚%{7}‘“
ftpquotatallies

//Definice třetí direktivy
QuotaTallyTable sql:/update-quota-tally
```

Čtvrtá direktiva je použita, pokud je založen účet novému uživateli a neexistuje záznam v tabulce záznamů. V tom případě je do tabulky vložen nový záznam.

```
SQLNamedQuery insert-quota-tally
INSERT „%{0},%{1},%{2},%{3}, %{4}, %{5}, %{6}, %{7}“
ftpquotatallies

//Definice čtcrté direktivy
QuotaTallyTable sql:/insert-quota-tally
```

Aby uživatelé měli přehled o dostupném prostoru na jejich účtu, předáme tyto informace pomocí protokolu FTP jejich klientovi.

```
//SQL dotaz pro zjištění kolik prostoru má uživatel
obsazeno //v MB
SQLNamedQuery gettally SELECT „
ROUND((bytes_in_used/1048576),2)
FROM     ftpquotatallies WHERE name=‘%u‘“

//SQL dotaz pro zjištění velikosti uživatelova prostoru v MB
SQLNamedQuery getlimit SELECT
„ROUND((bytes_in_avail/1048576),2)
FROM     ftpquotalimits
WHERE    name=‘%u‘“

//SQL dotaz pro zjištění zbývajícího místa v MB
SQLNamedQuery getfree SELECT „
ROUND(((
         ftpquotalimits.bytes_in_avail-
         ftpquotatallies.bytes_in_used)/2^10),2)
FROM     ftpquotalimits,ftpquotatallies
WHERE    ftpquotalimits.name = ,%u‘ AND
         ftpquotatallies.name = ,%u‘“

//Předání získaných parametrů ftpklientovi
SQLShowInfo LIST „226“
„Used %{gettally}MB from %{getlimit}MB.
You have %{getfree}MB available space.“
```

Po nainstalování a nakonfigurování Proftpd je nutné server spustit a otestovat funkčnost připojení k serveru. Nejdříve musíme připravit v systému složku, kde budou umístěny domény a jednotlivé složky uživatelů a ošetříme jejich práva.

Vytvoříme testovací účet v databázi spustíme server a otestujeme možnost připojení se k ftp serveru :

```
INSERTINTO`hosting`.`hostuser`
(`id_user`,`login`,`passwd`,`uid`,`gid`,`homedir`,`shell`)
VALUES
(,‘, ,test‘, ,test‘, ,2000‘, ,2000‘, ,/home/‘, ,/bin/false‘)
```

```
# /etc/init.d/proftpd start
```

Pokud start serveru proběhne bez chybových hlášek a k serveru se nám podaří připojit, nastavíme systém tak, aby Proftpd nastartoval automaticky po spuštění systému. To je důležité například při výpadku elektřiny. Po obnovení dodávky energie tak dojde k automatickému spuštění potřebných služeb.

```
# rc-update add /etc/init.d/proftpd default
```

Tímto příkazem zajistíme automatické spuštění služby v daném run-levelu[31] systému. Pokud se spuštění Proftpd nepodařilo nebo se k serveru i po zadání správných přihlašovacích údajů nedaří přihlásit, spustíme Proftpd ručně v ladícím režimu, kdy jsou hlášky přímo vypisovány na konzoli, což administrátorovi pomůže zjistit příčinu nefunkčnosti serveru.

31 Linuxový systém může běžet v různých režimech tzv. run-levelech. Run-levely jsou mimo jiné rozlišeny službami, které jsou v konkrétním režimu spuštěny.

2.2 Webový server

Další nedílnou součástí webhostingového serveru je http server. Pomocí webového serveru je pro uživatele zajištěno hned několik služeb. Zaprvé je pomocí něj možné přistupovat k webové prezentaci jednotlivých uživatelů. Dále zajišťuje přístup do emailových schránek uživatelů přes webové rozhraní a stejným způsobem umožňuje uživatelům přistupovat do databáze MySQL, pokud jí mají aktivovanou. Poslední funkce, kterou server zajišťuje, je webová prezentace firmy, která server provozuje. Tato část obsahuje také rozhraní, přes které je možné jako administrátor ovládat jednotlivé části serveru a administraci uživatelů. Uživatelé zde můžou spravovat své účty.

Jako zástupce http serveru z řad Opensource jsem jednoznačně zvolil produkt Apache od seskupení Apache Group. Vývoj tohoto software začal v roce 1993 ve společnosti NCSA. Dnes se na jeho vývoji podílí programátoři a administrátoři z celého světa.

2.2.1 Instalace Apache do operačního systému Linux Gentoo

Server Apache nainstalujeme do systému pomocí emerge. Před instalací musíme nastavit příznak SSL do proměnné USE, čímž se Apache zkompiluje s podporou zabezpečeného připojení SSL[32].

```
# emerge apache
# rc-update add /etc/init.d/apache2 default
```

Po nainstalování Apache program hledá své konfigurační soubory v adresáři /etc/apache2. Hlavní konfigurační soubor je httpd.conf. Po instalaci Apache zajistíme jeho automatické spuštění po startu systému.

2.2.2 Instalace web-mailového rozhraní Roundcubemail

Přístup k emailovým složkám uživatelů a jejich emailům je možné zajistit několika způsoby. Nejstarší a nejjednodušší z nich je přístup do mailové schránky po přihlášení uživatele do systému a následné vyzvednutí pošty některým z nainstalovaných klientů jako jsou **pine** nebo **mutt**. Tento způsob je dnes již považován za zastaralý, avšak v některých případech má své opodstatnění či jednoduše někomu vyhovuje pro své „kouzlo“. Pro účely webhostingového serveru je tento způsob nepoužitelný už z toho důvodu, že virtuální uživatelé nemají možnost přihlášení k systému pomocí programu Telnet či Ssh[33].

Další možností je použití některého z emailových klientů, např. Mozilla Thunderbird nebo Outlook od společnosti Microsoft. Tyto programy používají pro připojení k uživatelově schránce

32 **S**ecurity **S**ocket **L**ayer je protokol, resp. vrstva vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP), která poskytuje zabezpečení komunikace šifrováním a autentizaci komunikujících stran.

33 **Ssh** - Protokoly umožňující připojení k Unixovému stroji přes počítačovou síť. Starší protokol Telnet přenáší veškerou komunikaci nešifrovaně včetně přístupových hesel a jeho použití v dnešní době se nedoporučuje. Novější Ssh(existují dvě verze) šifruje spojení pomocí asymetrických šifer RSA nebo DSA.

protokol imap nebo pop3. Použití těchto programů je vhodné především na počítači, který uživatel často používá. Nastavení programu vyžaduje od uživatele jisté znalosti problematiky.
Poslední způsob přihlášení je pomocí www rozhraní, ve kterém jsou veškeré parametry nastaveny a jediné, co uživatel potřebuje znát, jsou jeho přihlašovací údaje. Opensource www rozhraní existuje opět celá řada, např. Horde, Roundcubemail, Squiremail, OpenWebMail a další.

Z těchto zástupců jsem zvolil poměrně nový software Roundcubemail. Je to jednoduché rozhraní s příjemným vzhledem a snadnou konfigurací. Je napsaný v Php a využívá funkce AJAX[34]. Ke své činnosti potřebuje mít přístup k databázi MySQL nebo PostgreSQL, kde si uchovává údaje o uživatelích. Dále Roundcubemail potřebuje funkční IMAP server, přes který se připojuje do emailových schránek uživatelů a výstup zobrazuje uživateli pomocí generování www stránek.

Pro instalaci je třeba získat zdrojové kódy programu. Ty je potřeba pouze extrahovat z archivu do adresáře, kde jej očekává server webový server Apache. Konfigurační soubory Roundcubemail jsou umístěny v adresáři config, který je součástí kořenového adresáře programu. Složka obsahuje soubory db.inc.php, ten slouží pro nastavení přístupu Roundcubemail k databázi MySQL respektive PostgreSQL. Soubor main.inc.php je hlavní konfigurační soubor.

Stažené zdrojové kódy web-mailového klienta Roundcubemail, které můžeme nalézt například na stránkách výrobce http://www.roundcube.net/downloads nejdříve nakopírujeme do cílové složky a extrahujeme.

```
# tar –xvjzf roundcubemail-0.1-rc2.tar.gz
```

Roundcubemail potřebuje pro svoji činnost přístup do databáze MySQL a proto je potřeba pro něj vytvořit uživatele a databázi, do které bude mít přístup. Roundcubemail také vyžaduje tabulky, které však není potřeba vytvářet ručně, protože jejich struktura byla tvůrci vyexportována a přiložena ke zdrojovým kódům. Tabulky proto pouze importujeme do MySQL.

2.2.3 Vytvoření nového uživatele pro Roundcubemail v MySQL

```
# mysql –u root –p
    mysql> create database roundcube;
    mysql> grant ALL PRIVILEGES ON roundcube.* TO \
           roundcube@localhost IDENTIFIED BY ‚$password‘;
    mysql> flush privileges;
    exit;
```

34 **AJAX** - Asynchronous JavaScript and XML je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovu načítání.

2.2.4 Importování tabulek do MySQL

```
#mysql –u root –p roundcube < \
 home/www/roundcube/SQL/mysql.initial.sql
```

2.2.5 Konfigurace klienta Roundcubemail

Nastavení připojení Roundcubemail provedeme v souboru configure/db.inc.php

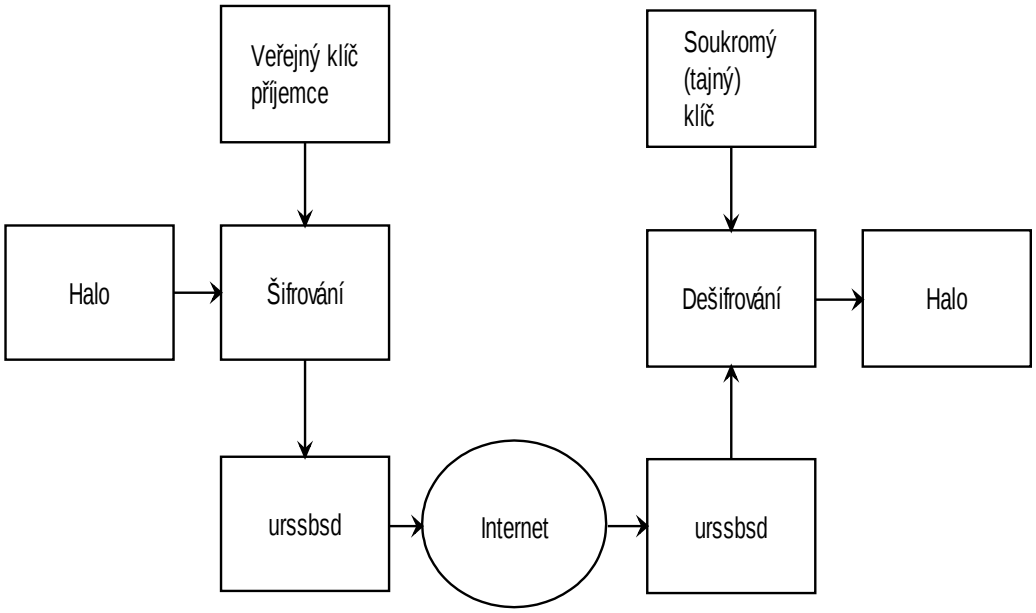
```
//Uživatel který bude použit pro připojení k mysql
$rcmail_config['db_dsnw'] =
'mysql://$password@localhost/roundcube';
```

Další nastavení provedeme v souboru configure/main.inc.php

```
/* Nastavení podle doporučení tvůrce programu pro případy,
kdy IMAP server je na stejném systému jako Roundcubemail */
$rcmail_config['enable_caching'] = FALSE;

/* Roundcube automaticky vytvoří záznam o uživateli ve své
databázi pokud je uživatel ověřen */
$rcmail_config['auto_create_user'] = TRUE;

//Jméno stroje na kterém Roundcubemail běží
$rcmail_config['default_host'] = 'mail.example.com';

/* Nastavení portu, který je používán při odesílání zpráv.
Protože server používá autentizaci při odesílání zpráv přes
SMTP,roundcube použije přihlašovací údaje uživatele pro SASL
přihlášení uživatele. */

$rcmail_config['smtp_port'] = 25;
$rcmail_config['smtp_user'] = '%u';
$rcmail_config['smtp_pass'] = '%p';
```

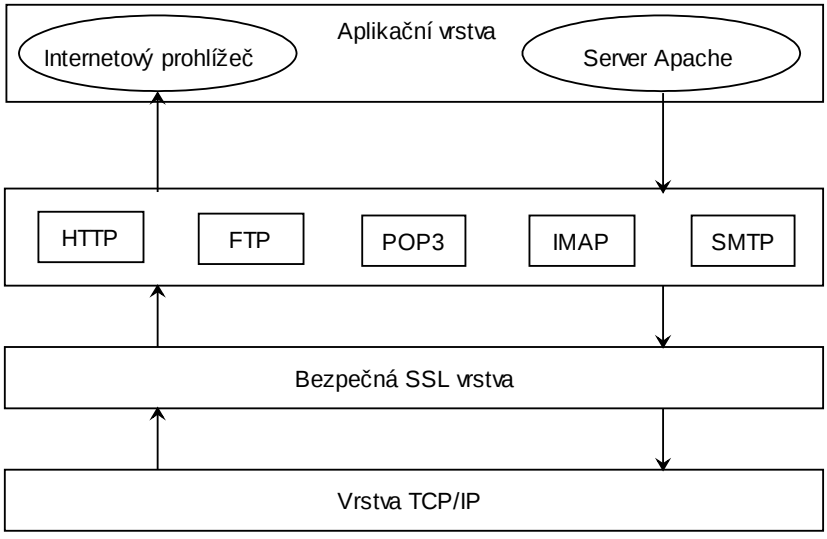
2.2.6 Principy SSL (Secure Socket Layer)

Základním principem SSL, jak je uvedeno v [8], je šifrování a SSL přímo definuje, jaký typ bude při zabezpečené komunikaci použit. SSL používá pro zašifrování dat symetrickou šifru a pro výměnu a ustanovení šifrovacích klíčů asymetrické šifrovací schéma s veřejným a privátním klíčem. Data, která jsou šifrována veřejným klíčem, mohou být poté dešifrována pouze soukromým klíčem. Schéma šifrování je zobrazeno na obr 2.1 a obecné schéma práce SSL je zobrazeno na obr. 2.2.

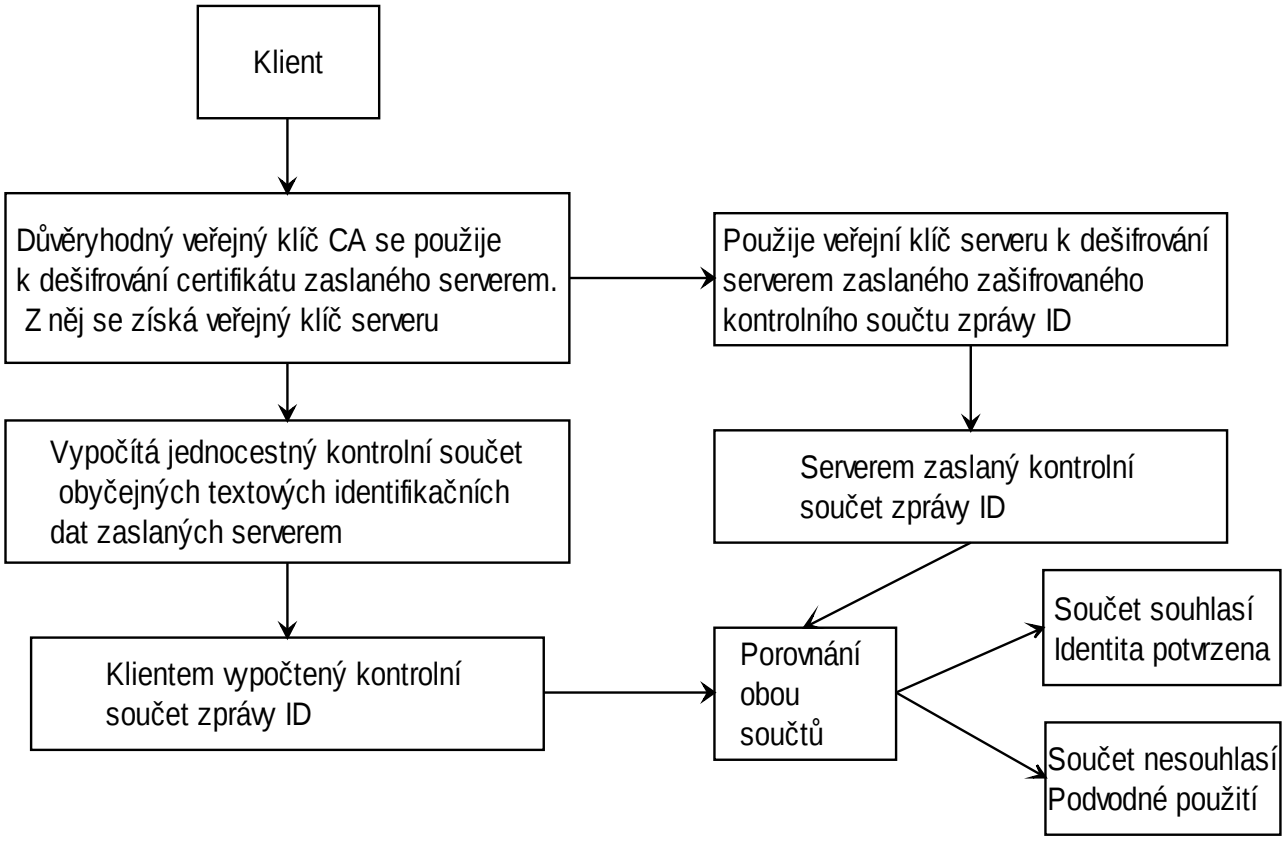


obr. 2.1. Princip asymetrického šifrování

Pro distribuci veřejného klíče a zároveň ověření pravé identity subjektu slouží Certifikáty. Ty obsahují informace uvedené v tabulce tab. 2.4. a dále mohou obsahovat informace např. dobu, pro kterou jsou platné, sériové číslo atd. Tyto základní údaje uvedené v tabulce, je potřeba před generováním certifikátu vyplnit. Princip SSL transakce podle [8] je znázorněn na obr. 2.3.



obr. 2.2. Základní schéma práce SSL



obr 2.3 Princip SSL transakce

Název pole	Zkratka	Význam	Nastaveno
Common Name	CN	Jméno subjektu	mail.maxprogres99.cz
Organization	O	Jméno organizace	Maxprogres
Organizational Unit	OU	Oddělení/Divize	-
City	L	Město	Brno
State/Province	ST	Stát	-
Country	C	Země	Czech Republic

Tab 2.4 Jmenná pole certifikátu

2.2.7 Vytvoření SSL certifikátu pro Apache pomocí OpenSSL

Apache potřebuje tyto certifikáty pro své virtuální servery, které budou provozovány zabezpečeně. Toto připojení je vhodné pro stránky, kde uživatelé zadávají svá uživatelské jména a hesla. Komunikace mezi serverem a webovým prohlížečem je potom šifrovaná, což znemožňuje případnému útočníkovi odposlechnout údaje. Uživatel si také pomocí vygenerovaného certifikátu může ověřit, že údaje nezadává do útočníkem zfalšované stránky, čímž zabráníme tzv. Man-in-the-middle útoku. Na tomto serveru bude tento typ spojení použit pro přístup k web-mailovému rozhraní Roundcubemail a to na adrese mail.maxprogres99.cz

Nejdříve je nutné do systému nainstalovat balík OpenSSL jak popisuje [6].

```
# emerge openssl
```

Před vygenerováním self-signed[35] certifikátu je potřeba vyplnit identifikační údaje vlastníka certifikátu v souboru /cd/ssl/openssl.cnf. Poté je možné vytvořit certifikát.

```
# openssl req -new > new.cert.csr
# openssl rsa -in privkey.pem -out new.cert.key
# openssl x509 -in new.cert.csr -out new.cert.cert -req \
          -signkey new.cert.key -days 365
```

Dále podle [6] certifikáty nakopírujeme do adresáře Apache

```
# cp /etc/ssl/misc/new.cert.cert /etc/apache2/ssl/
# cp /etc/ssl/misc/new.cert.key /etc/apache2/ssl/
```

Aby Apache mohl tyto certifikáty využívat, musíme vytvořit nový konfigurační soubor, kde nastavíme virtuální servery využívající SSL a cestu k certifikátům.

```
# touch /etc/apache2/vhosts/ssl-host.conf
```

Definice Virtuálního serveru mail.kolonoh.net, který naslouchá na portu 443. Tímto říkáme prohlížeči , že má používat zabezpečený protokol HTTPS[36] pro přenos dat.

Do souboru ssl-host.conf potom přidáme

```
<VirtualHost *:443>
```

Dále je potřeba zadefinovat jméno serveru a emailovou adresu správce serveru

```
ServerName mail.maxprogres99.cz
ServerAdmin admin@maxprogres99.cz
```

Domovský adresář serveru a pravidla pro tento adresář nastavíme pomocí těchto direktiv

```
DocumentRoot /home/www/roundcubemail
<Directory "/home/www/roundcubemail/">
</Directory>
```

Jako poslední je potřeba nastavit cestu k SSL certifikátům

```
SSLCertificateFile /etc/apache2/ssl/new.cert.cert
```

35 Self-signed certifikát je certifikát podepsaný sám sebou, naší vlastní certifikační autoritou

36 **HTTPS** je nadstavba počítačového protokolu HTTP, která poskytuje zvýšenou bezpečnost před odposloucháváním či podvržením dat.

```
SSLCertificateKeyFile /etc/apache2/ssl/new.cert.key
SSLEngine on
```

Pro takto nakonfigurovaný Apache může uživatel k web-mailovému rozhraní přistupovat, pokud do svého webového prohlížeče zadá celou URL[37] adresu a specifikuje port nebo protokol. To je v praxi nepoužitelné, protože většina uživatelů je zvyklá pouze zadat adresu a o použitém protokolu či portu nic netuší. Z toho důvodu je pro tyto případy zajistit přesměrování požadavku na správný port. To můžeme zajistit pomocí modulu Apache mod_rewrite, který slouží k přepisování URL požadavků dle zadaných pravidel. Pro zápis pravidel je využito regulárních výrazů.

Při zadání adresy mail.maxprogres99.cz do webového prohlížeče se odešle požadavek na port 80, kde standardně naslouchá http server. Webové rozhraní pro mail je však spojeno se zabezpečeným portem 443, na který je potřeba webový prohlížeč přesměrovat. Proto je potřeba server Apache nastavit tak, že pokud obdrží dotaz http://mail.maxprogres99.cz ,má ho přepsat na https://mail.maxprogres99.cz a teprve poté zpracovat.

Dále je třeba do hlavního konfiguračního souboru přidat virtuální server přiřazený k URL http://mail.maxprogres99.cz a v tomto kontejneru zajistit přepsání lokátoru.

Do konfiguračního souboru Apache httpd.conf přidáme tyto řádky

```
<VirtualHost *:80>
ServerName  mail.maxprogres99.cz
Serveradmin admin@maxprogres99.cz
/* Zapnutí modulu mod_rewrite a zapsání pravidla pro přepis
URL lokátoru. */
RewriteEngine On
RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [L,R]
</VirtualHost>
```

Restartujeme Apache pro znovu načtení konfiguračních souborů a otestujeme funkčnost pomocí webového prohlížeče. Všechny požadavky, které nyní směřují na adresu mail.maxprogres99.cz, jsou přesměrovány na zabezpečený port 443. Podrobný popis modulu Rewrite je možné nalézt v [8]

2.2.8 Dynamická konfigurace virtuálních serverů pro Apache

Pojem virtuálních serverů ve spojení s Apache používáme v případech, kdy je provozováno více webových serverů na jednom stroji. Virtuální servery mohou být rozlišeny podle IP adres či podle názvů web stránek. Pro práci s virtuálními hosty Apache potřebuje modul mod_vhost .

37 **URL**(Unique Resource Locator) znamená jednoznačné určení zdroje. Je to způsob, jak jednoznačně zapsat umístění souboru na Internetu nebo na intranetu.

Jednotlivé virtuální servery jsou potom uzavřeny v kontejneru pomocí direktivy <VirtualHost></Virtualhost>. Kontejner dále obsahuje položky ServerName, ServerAdmin a další, jejichž význam je možné nalézt v dokumentaci Apache a v [8].

Na serveru poskytující mwebhostingové služby se předpokládá velký počet virtuálních hostů a proto je potřeba vyřešit jejich správu. Ruční přidávání jednotlivých virtuálních serverů do konfiguračního serveru je možné maximálně do několika desítek takových serverů. Jediný server Apache však v závislosti na výkonu stroje dokáže obsloužit i 1000 virtuálních serverů na jednom stroji. Přicházelo by v úvahu vytvořit například Perlový[38] skript, který by automaticky generoval soubor s virtuálními hosty. To má ovšem dvě velké nevýhody. Při každém přidání nebo odebrání uživatele by bylo třeba Apache restartovat, aby načetl změny v konfiguračním souboru, což by bylo v praxi nepoužitelné. Velký počet virtuálních umožňuje při vytváření cest k fyzickým adresářům potřebným k obsluze webu použít adresu IP nebo hostitelské jméno konfigurace virtuálních hostitelů.

Kořenový adresář dokumentů pro virtuálního hostitele umožňuje nastavit direktiva VirtualDocumentRoot pomocí interpolované adresářové cesty. Při použití tohoto modulu a direktivy VirtualDocumentRoot je potřeba nastavit UseCanonicalName na hodnotu Off, čímž bude Apache při získávání jména hostitele používat hlavičku Host. Direktiva VirtualDocumentHost je vhodná v případech, kdy máme jednu IP adresu odpovědnou za mnoho virtuální webových serverů. Velké množství virtuální kontejnerů také způsobí, že Apache dlouho startuje a zabírá velké množství paměti.

Server Apache proto nabízí možnost nakonfigurovat virtuální hosty dynamicky pomocí modulu mod_vhost_alias jak popisuje [8]. Tento modul má význam pouze u instalací Apache, kde se předpokládá velké množství virtuálních hostitelů. Modul umožňuje při vytváření cest k fyzickým adresářům potřebným k obsluze webu použít adresu IP nebo hostitelské jméno konfigurace virtuálních hostitelů.

Kořenový adresář dokumentů pro virtuálního hostitele umožňuje nastavit direktiva VirtualDocumentRoot pomocí interpolované adresářové cesty. Při použít tohoto modulu a direktivy VirtualDocumentRoot je potřeba nastavit UseCanonicalName na hodnotu Off, čímž bude Apache při získávání svého jména používat hlavičku Host, kterou získá od prohlížeče. Direktiva VirtualDocumentHost je vhodná v případech, kdy máme jednu IP adresu odpovědnou za mnoho virtuálních webových serverů.

Příklad použití

```
UseCanonicalName        off
VirtualDocumentName     /www/%0/htdocs
```

38 Perl je interpretovaný programovací jazyk, určený především pro tvorbu CGI skriptů.

V tomto případě je hodnota 0% nahrazena úplným jménem hostitele, např. tedy www.maxprogres99.cz. Takovéto použití dynamického určování virtuálního hostitelského adresáře by však bylo příliš hrubé a jednotlivé weby by bylo lepší rozlišovat nejprve podle domény, pod kterou spadají a poté podle jejich přiděleného uživatelského jména. K tomu můžeme využít pro získání doménové části adresy 2%+, což v našem příkladě znamená část maxprogres99.cz a pro získání části uživatele %1 vrátí část www.

Pro nakonfigurování Apache podle tohoto scénáře je potřeba do souboru http.conf přidat tyto řádky.

```
UseCanonicalName off
<VirtualHost *:80>
    ServerAdmin admin@kolonoh.net
    VirtualDocumentRoot /home/www/%2+/%1/
    ServerName www.kolonoh.net
</VirtualHost>
```

2.2.8 Instalace webového rozhraní pro MySQL - phpMyAdmin

phpMyAdmin je jeden z nejlepších nástrojů pro správu databází MySQL. Jak je patrné již z názvu, program je postaven na Php. Poskytuje uživateli prostředek, kterým může přistupovat k databázi provozované na hostingovém serveru. Běžný uživatel většinou práci v příkazovém řádku s databází MySQL neovládá a jak již bylo uvedeno, přihlašování uživatelů, kteří využívají služby hostingového serveru, přímo k unixovému terminálu není povoleno. Správa databáze v phpMyAdmin je velice přehledná, jednoduchá a intuitivní. Databázi pro své webové stránky tak může využívat i naprostý nováček, což zvyšuje atraktivitu poskytovaných služeb.

Autentizace v phpMyAdmin se provádí vůči MySQL a to třemi možnými způsoby. Výběr způsobu se provádí v konfiguračním souboru phpMyAdmin, kde je na výběr ze třech možností.

Nejdříve je nutné phpMyAdmin získat pomocí nástroje emerge a nainstalovat do systému.

```
# emerge phpmyadmin
```

Dále je třeba nakonfigurovat MySQL, tj. přidat uživatele phpMyAdmin, který pro svojí činnost vyžaduje uživatele s právy pro čtení tabulek přístupů.

```
# mysql -u root -D mysql -p
mysql> GRANT USAGE ON mysql.* TO 'phpmyadmin'@'localhost' \
IDENTIFIED BY '$password';

mysql> GRANT SELECT (Host, User, Select_priv, Insert_priv, \
 Update_priv, Delete_priv, Create_priv, Drop_priv, \
 Reload_priv, Shutdown_priv, Process_priv, \
 File_priv, Grant_priv, References_priv, Index_priv, \
 Alter_priv, \
```

```
 Show_db_priv, Super_priv, Create_tmp_table_priv,\
 Lock_tables_priv, \
 Execute_priv, Repl_slave_priv, Repl_client_priv \
       ) ON mysql.user TO 'phpmyadmin'@'localhost';

mysql>GRANT SELECT ON mysql.db TO 'phpmyadmin'@'localhost';

mysql>GRANT SELECT ON mysql.host TO'phpmyadmin'@'localhost';

mysql>GRANT SELECT (Host, Db, User, Table_name, Table_priv,\
     Column_priv) ON mysql.tables_priv TO 'phpmyadmin'@ \
    'localhost';
```

Abychom mohli využívat všechny funkce phpMyAdmin, musíme pro tento program vytvořit samostatnou databázi, se kterou bude pracovat a nastavit příslušná práva.

```
mysql> create database phpmyadmin;
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON phpmyadmin.*
TO phpmyadmin'@' localhost';
```

Tím je hotova konfigurace MySQL a jako poslední krok je potřeba přidat virtuální server pro phpMyAdmin do webového serveru Apache.

```
<VirtualHost *:80>
        ServerName      mysql.maxprogres99.cz
        ServerAlias     mysql.*.*
        DocumentRoot    /home/www/phpmyadmin/
</VirtualHost>
```

Po restartování Apache bude phpMyAdmin přístupný na adrese http://mysql.maxprogres99.cz , kde se uživatel po zadání přihlašovacích údajů dostane ke své databázi.

2.2.9 Statistiky návštěvnosti www serveru

Důležitou informací pro každého provozovatele internetových stránek je informace o tom, jak jsou jejich stránky navštěvované. Na základě těchto informací může provozovatel upravit obsah stránek, marketingovou strategii atd. V případě reklamních bannerů je návštěvnost stránek důležitým ukazatelem při určování ceny za poskytnutý prostor. Statistiky o návštěvnosti internetových stránek je možné získat z logovacích souborů serveru Apache.

Existuje několik nástrojů, které z těchto záznamů dokáží udělat přehledné statistiky rozdělené podle data. Z aplikací, které umožňují takové statistiky generovat, můžeme jmenovat např. **AWStats** nebo **Webalizer**. Na tomto hostingovém serveru jsem použil druhý zmíněný. Protože hostingový server poskytuje služby pro více domén, je potřeba statistiky návštěvnosti

vygenerovat pro každého zákazníka (tedy pro každou doménu). Toho docílíme tím, že ze záznamů serveru Apache vybereme pouze záznamy týkající se konkrétní domény a statistiky vygenerujeme pro tyto separované záznamy zvlášť a umístíme je např. do adresáře /graphs v kořenovém adresáři pro konkrétní doménu. Při generování statistik je možné programu Webalizer předat parametr, který určí vstupní soubor se záznamy přístupů na server a dalším parametrem určíme výstupní adresář pro vygenerované statistiky.

V základním nastavení Webalizer při generování statistik používá pomocný soubor, ve kterém jsou zaznamenány předešlé statistiky. Tento soubor umožňuje zrychlení zpracování statistik, protože není potřeba opakovaně vyhodnocovat starší záznamy. V našem případě je však použití tohoto souboru nevhodné, protože vyhodnocujeme záznamy pro různé domény. Použití souboru zamezíme příslušným parametrem.

Jak už bylo uvedeno výše, předtím než začneme statistiky návštěvnosti internetových stránek vyhodnocovat, je potřeba na základě databáze uživatelů tyto statistiky roztřídit. To můžeme zajistit např. skriptem napsaným v jazyce **BASH**[39]. Tento skript nejdříve z databáze získá seznam všech poskytovaných domén a na základě těchto znalostí potom provede separaci záznamů a vygenerování statistik. Tento skript je potřeba spouštět např. každý den v nočních hodinách, kdy není server tolik vytížen. Spouštění zajistíme např. pomocí nástroje Cron.

Webalizer je dostupný ve stromu Portage a můžeme jej nainstalovat do systému pomocí emerge.

```
# emerge webalizer
```

Dále je potřeba vytvořit soubor, který zajistí separaci a generování statistik.

```
#!/bin/bash

/* Získání všech všech domén, pro které se budou statistiky
generovat */
for row in
  $(mysql -N -D hosting -u root --password="secret"
                        -e 'select login from hostuser')
    do(

// získání uživatelského jména
        jmeno=`echo $row | awk -F . '{print $1}'`

// získání doménové části
        domena=`echo $row | awk -F . '{print $2"."$3}'`

// vytvoření cesty ke kořenovému adresáři virtuálního serveru
        cesta="/home/www/$domena/$jmeno"
```

39 **BASH** (Bourne Again SHell) **,** je prostředí příkazové řádky, ale zároveň skriptovací jazyk

```
// dočasný adresář pro generování statistik
        tmp="/var/tmp/httpd/$jmeno/$jmeno"

/* je potřeba zjistit, zda existuje adresář pro uložení
vygenerovaných statistik */
        if test -d $cesta/graphs
           then echo Uz existuje > /dev/null;
           else mkdir $cesta/graphs
            fi

// vyseparování statistik za poslední dva dny
        cat /var/log/httpd/access_log | grep "$row" > $tmp
        cat /var/log/httpd/access_log.1 | grep "$row" > $tmp
//vygenerování a uložení statistik pro jednotlivé domény
        webalizer -i -t $row -o $cesta/graphs $tmp
        )
    done
```

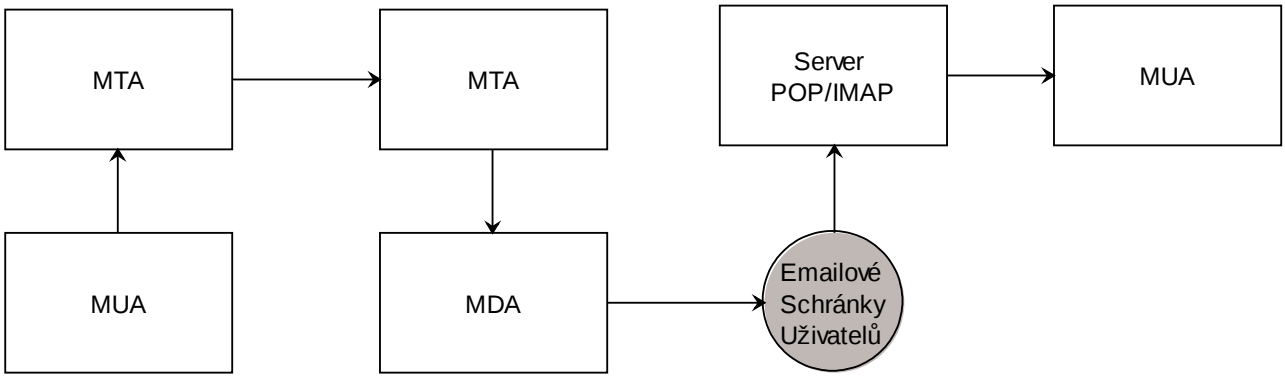
Tento zdrojový kód uložíme do souboru, a zajistíme jeho pravidelné spouštění pomocí programu Cron v nočních hodinách, kdy není server tolik zatížen.

2.3 Poštovní server

2.3.1. Úvod do poštovních serverů

Počátek elektronické pošty, jak uvádí [3], je datován již od počátků 70 let, kdy sloužila pro odesílání poštovních zpráv síť Arpanet[40], předchůdce dnešního internetu. Od té doby se elektronická pošta stala nejvyužívanější službou na internetu. Protože internet je velmi heterogenní síť, je zapotřebí pro přesun pošty velice flexibilní nástroj. Tím se stal na počátku 80. let balíček Sendmail, který velmi rychle získal dominantní postavení na poli poštovních serverů. Jeho konfigurace a správa je však velmi obtížná a Sendmail také velmi brzy začal mít velké problémy se zabezpečením. Průměrně byla objevena jedna bezpečností slabina v Sendmailu každý týden, což bylo způsobeno zejména jeho monolitickou architekturou.

Kvůli velkým problémům se serverem Sendmail začaly v pozdějších dobách vznikat náhrady za tento software. Mezi ty nejznámější patří trojice poštovních serverů Postfix, Qmail a Exim. Postfix napsal Wietse Venema a jeho zdrojové kódy byly uvolněny v roce 1998. Poštovní server postfix byl od počátku tvořen tak, aby nahradil převládající Sendmail. Je koncipován tak, aby se co možná nejlépe vyrovnal s případnými neočekávanými problémy, které mohou nastat. Hlavní znaky postfixu jsou spolehlivost, zabezpečení, výkon, flexibilita, snadné používání a kompatibilita se Sendmailem jak je uvedeno v [3].



obr. 2.4 Obecný popis komunikace SMTP

2.3.2 Instalace poštovního server postfix do operačního systému Gentoo

Postfix je samozřejmě standardní součástí Portage stromu a pro jeho instalaci tedy použijeme nástroj emerge, jak je popsáno v [6]. Nejprve je však třeba nastavit proměnnou USE, aby se Postfix zkompiloval se správnými parametry. Po nainstalování a spuštění Postfixu zajistíme, aby se služba spouštěla automaticky po startu systému.

40 **Arpanet**(Advanced Research Projects Agency Network) – armádní počítačová síť, která vznikla v roce 1969 a stala se zárodkem dnešního intertnetu.

```
# emerge postfix
# /etc/init.d/postfix start
# rc-update /etc/init.d/postix default
```

2.3.3. Nastavení Postfixu pro hostování více domén a spolupráci s MySQL

Postfix může podle [3] zpracovávat poštu pro virtuální domény čtyřmi způsoby

- sdílené domény se systémovými účty
- oddělené domény se systémovými účty
- oddělené domény s virtuálními účty
- virtuální domény s vlastním úložištěm, které není spravováno postfixem

Pro konfiguraci sdílených domény se systémovými účty přijímá každý uživatel zprávy pro každou doménu. Informace o tom, pro jaké virtuální domény má server poštu přijímat postfix, získá z parametru $mydestination v hlavním konfiguračním souboru postfixu main.cf. Tato konfigurace je nejjednodušší, avšak pro naše použití není vhodná z výše uvedených důvodů.

V případě oddělených domén je pošta určená pro různé domény doručována různým uživatelům v systému. Při této konfiguraci je zapotřebí použít parametry virtual_alias_domains pro nastavení virtuálních domén a parametr virtual_alias_maps, který se postará o namapování emailových adres na konkrétní uživatelské účty. Toto nastavení je pro použití na hostingovém serveru vyhovující z hlediska oddělení domén, ale potřeba mít pro každého uživatele systémový účet je nevyhovující.

Jak bylo uvedeno v předchozím textu, nevýhoda obou předchozích technik je to, že je potřeba spravovat systémové účty pro všechny emailové adresy na serveru. Pokud však Postfix nastavíme tak, že bude používat oddělené domény s virtuálními účty. Postfix poté zprávy doručuje do místního úložiště , kde má každá virtuální emailová adresa svůj vlastní soubor schránky. Uživatelé pak své zprávy přijímají prostřednictvím vhodně nastaveného POP/IMAP serveru. Toto doručování funguje jako běžné místní doručování, avšak odpadá potřeba mít pro každý emailový účet odpovídající uživatelský účet. Nejprve je zapotřebí Postfixu sdělit, aby pro doručování zpráv používal tabulky s virtuálními záznamy. To provedeme pomocí parametru **virtual_transport**, který nastavíme na hodnotu virtual jak uvádí i [14]. Takto nastavený server potom přijímá zprávy určené pro domény určené parametrem **virtual_mailbox_domains**. Dále je potřeba pomocí parametru **virtual_mailbox_base** nastavit základní adresář úložiště zpráv. Jednotlivé emailové adresy je třeba namapovat na soubory schránek uživatelů. Nastavení se provádí parametrem **virtual_mailbox_maps**. Každý uživatel přijímající poštu potom musí mít odpovídající záznam ve vyhledávací tabulce Postfixu daného typu. Umístění konkrétní schránky je bráno jako relativní

cesta oproti parametru **virtual_mailbox_base**. Postfix poskytuje opravdu velké množství parametrů, kterými lze jemně nastavovat chování celého systému. Např. parametr **virtual_alias_maps** umožňuje nastavení aliasů pro uživatele. Tyto aliasy jsou samozřejmě uloženy v databázi. Seznam všech parametrů i s jejich popisem je uveden v [14]

Jak uvádí [14], Postfix dokáže jako vyhledávací tabulky využívat i databázové formáty a dokáže spolupracovat i celým databázovými systémy např. MySQL nebo LDAP. Protože tento hostingový server má většinu informací o uživatelích již uloženou v databázi MySQL, bude výhodné využít spolupráce Postfixu a MySQL, i když výkon postfixu bude tímto o něco snížen.

Při konfiguraci Postfixu s databází MySQL v konfiguračním souboru main.cf je podle [14] potřeba nastavit druh mapy a soubor, který obsahuje mapování. Do souborů s mapováním musíme zadat informace, jak získat příslušné hodnoty z databáze.

Pro nastavení spolupráce Postfixu s MySQL v konfiguračním adresáři /etc/postfix vytvoříme pro přehlednost adresář mysql, do kterého umístíme soubory s mapováním a mapování nastavíme v konfiguračním souboru postfixu. Tabulka v databázi MySQL, která obsahuje informace o emailových účtech, se jmenuje mailsql.

Do souboru main.cf přidáme tyto řádky

```
// statický seznam hostovaných domén
virtual_mailbox_domains = maxprogres99.cz

// minimální uid virtuálního uživatele
virtual_minimum_uid     = 1000
```

```
/* informace o poštovních účtech jsou uložené v databázi
MySql */
virtual_mailbox_maps    = mysql:/etc/postfix/mysql/virtual-
                          maps.cf

virtual_mailbox_base    = /

virtual_alias_maps      = mysql:/etc/postfix/mysql/virtual-
                          alias.cf

//základní velikost poštovní schránky
virtual_mailbox_limit   = 20000
```

Výpis souborů obsahující parametry pro přístup k databázi

```
// virtual-maps.cf

user                    = mailsql
password                = $password
dbname                  = mailsql
```

```
table                   = users
select_field            = maildir
where_field             = email
additional_conditions   = and postfix = 'y'
hosts                   = unix:/var/run/mysqld/mysqld.sock

// mysql-virtual.cf

user                    = mailsql
password                = $password
dbname                  = mailsql
table                   = virtual
select_field            = destination
where_field             = email
hosts                   = unix:/var/run/mysqld/mysqld.sock
```

Sloupec	Typ	Význam	Výchozí
id	int(11)	Unikátní id emailové schránky	-
email	varchar(128)	Emailová adresa schránky	-
crypt	varchar(128)	Heslo v šifrované podobě	-
name	tinytext	Jméno vlastníka schránky	-
uid	int(11)	Systémové UID vlastníka schránky	1002
gid	int(11)	Systémové GID vlastníka schránky	1002
maildir	tinytext	Adresář pro uložení emailové schránky	-
quota	tinytext	Velikost emailové schránky	-
postfix	enum('n', 'y')	Určení zda je schránka aktivní/neaktivní	y

Tab 2.5 Struktura tabulky mailsql

2.3.4 Instalace Courier-IMAP

Po nainstalování a správném nakonfigurování poštovního serveru Postfix máme zajištěné doručování příchozích emailů do schránek uživatelů podle databáze MySQL. Dále je potřeba zajistit přístup jednotlivým uživatelům do jejich emailových schránek. Pro přístup do schránek uživatelů se využívají protokoly POP a IMAP. Tyto protokoly používají při přístupu k emailům různou filozofii, popisovat jednotlivé rozdíly mezi nimi je nad rámec toho textu. Důležité je, že server Courier-IMAP podporuje oba typy protokolů. Courier-IMAP podle [6] bude vyžadovat pomocnou součást Courier-authlib, která umožní ověřovat uživatele přistupující k jejich emailové schránce oproti databázi MySQL. Pro větší bezpečnost můžeme využít služeb SSL, a veškerá komunikace i přenos emailů budou zabezpečeny. Pro tento účel je třeba vygenerovat certifikáty. Balíček Courier-IMAP pro tento účel nabízí vlastní programy **mkpop3dcert** a **mkimapdcert**. Veškerý software uvedený v této kapitole je dostupný přes strom Portage.

```
# emerge courier-imap
# emerge courier-authlib
```

Po zkompilování a nainstalování balíčků musíme Courier nakonfigurovat. Konfigurační soubory jsou v adresáři /etc/Courier-IMAP. Podle [6] je pro vygenerování certifikátů třeba vyplnit soubory pop3d.cnf a imapd.cnf, které budou využity pro generování SSL certifikátů. Po vyplnění konkrétních údajů vygenerujeme pomocí dříve uvedených programů SSL certifikáty a spustíme jednotlivé servery. Samozřejmě zajistíme jejich spuštění při startu systému.

```
# mkpop3dcert
# mkimamdcert
# /etc/init.d/courier-imapd-ssl start
# /etc/init.d/courier-pop3d-ssl start
# rc-update /etc/init.d/courier-imapd-ssl default
# rc-update /etc/init.d/courier-pop3d-ssl default
```

Pokud servery nastartují v pořádku, je možné se k nim připojit např. pomocí programu Telnet, tím otestujeme zda servery naslouchají na svých portech. V tuto chvíli se však nemůžeme přihlásit k žádnému emailovému účtu, protože nebyl nastavena autentizační část Courier.

Konfigurační soubory pro autentizační část Courier se nacházejí v adresáři /etc/courier/authlib. Adresář obsahuje dva konfigurační soubory. V souboru authdaemonrc se nastavují moduly, které se používají při ověřování uživatelů. V případě kdy jsou uživatelé ověřováni vůči MySQL, použijeme modul authmysql , jak uvádí [6].

Výpis souboru authdaemonrc :

```
authmodulelist=“authmysql“
```

Druhý konfigurační soubor se jmenuje authmysqlrc, a obsahuje parametry, které ke své činnosti potřebuje autentizační modul authmysql. Mezi parametry patří např. informace o připojení k MySQL, informace o tabulce obsahující údaje o uživatelích a další.

Výpis souboru authmysqlrc :

```
MYSQL_SERVER            localhost
/* Uživatel mailsql v MySql, kterého musíme nejprve vytvořit.
Viz. předchozí kapitoly */
MYSQL_USERNAME          mailsql
MYSQL_PASSWORD          $password
MYSQL_DATABASE          mailsql
//Tabulka, která obsahuje informace o uživatelích
MYSQL_USER_TABLE        users
//Hesla jsou uložena v šifrované podobě
MYSQL_CRYPT_PWFIELD     crypt
```

```
//Názvy jednotlivých sloupců obsahujících potřebné údaje
MYSQL_UID_FIELD         uid
MYSQL_GID_FIELD         gid
MYSQL_LOGIN_FIELD       email
MYSQL_NAME_FIELD        name
MYSQL_MAILDIR_FIELD     maildir
```

Po dokončení těchto kroků můžeme autentizačního démona obvyklým způsobem spustit.

```
# /etc/init.d/courier-authlib restart
# rc-update add /etc/init.d/courier-authlib default
```

2.3.5 Instalace Cyrus-SASL

Po dokončení instalace serverů Postfix a Courier-IMAP je server schopný přijímat emaily a umožňuje uživatelům přístup do jejich emailové schránky. Dále umožníme uživatelům posílat emailové zprávy skrz tento poštovní systém. Předtím, než je uživateli umožněno emailovou zprávu odeslat, je potřeba uživatele ověřit. Tím zamezíme zneužití poštovního systému k rozesílání nevyžádané pošty a zároveň tím chráníme samotné uživatele a jejich poštovní účty. Pro ověřování identity uživatele využívající protokol SMTP[41] používá Postfix Cyrus- SASL[42], který zajistí ověření uživatele v databázi MySQL pomocí již nainstalované knihovny Courier-authlib. Po nainstalování balíčku Cyrus musíme podle [6] nastavit systém tak, aby při obsluze SMTP komunikace používal právě jeho služby. A nakonec musíme nakonfigurovat i samotný Cyrus, aby k autentizaci využíval služeb Imap serveru. Konfiguraci služby smtpd najdeme v souboru /etc/sasl2/smtpd.conf a nastavení parametrů pro autentizaci provedeme v souboru /etc/conf.d/saslauthd.

```
#emerge cyrus-sasl
```

Soubor /etc/sasl2/smtpd.conf

```
// seznam metod používaných při autentizaci
mech_list = PLAIN LOGIN
// metoda získání údajů o uživateli
pwcheck_method = saslauthd
```

Soubor /etc/conf.d/saslauthd

```
/* Abychom mohli využít služeb knihovny courier-authlib,
předáme přihlašovací údaje serveru IMAP, který pro Cyrus
provede požadované ověření */
SASLAUTHD_OPTS="${SASLAUTH_MECH} -a rimap -r"
```

41 **SMTP**(Simple Mail Transfer Protocol) – Protokol pro přenos emailových zpráv přes heterogenní sítě. Používá se pro přenos zpráv od klienta na poštovní server a při komunikaci mezi poštovními servery.

42 **SASL**(Simple Authentication and Security Layer) **–** Vrstva zajišťující autentizaci spojově orientovaných protokolů.

```
// Jako adresa serveru IMAP nastavíme hodnotu localhost
SASLAUTHD_OPTS="${SASLAUTHD_OPTS} -O localhost"
```

Více informací o parametrech příkazu saslauthd je možné získat pomocí manuálových stránek (příkaz man saslauthd). Službu spustíme a zajistíme její automatické spouštění.

```
# /etc/init.d/saslauthd start
# rc-update add /etc/init.d/saslauthd default
```

2.3.6 Konfigurace postfixu pro podporu SASL a TLS

Jak je uvedeno v [6], je po nainstalování Courier-IMAP a Cyrus-SASL třeba Postfix nakonfigurovat tak, aby tyto služby využíval. Veškeré nastavení se provádějí v hlavním konfiguračním souboru Postfixu main.cf. Pro aktivaci ověřování klientů – SASL je podle [6][14] potřeba do konfiguračního souboru přidat následující řádky :

```
// v souboru main.cf
// Nastaví Postfix tak aby používal Sasl autentizaci
smtpd_sasl_auth_enable = yes
smtpd_sasl2_auth_enable = yes

// Zakázání anonymního přihlašování
smtpd_sasl_security_options = noanonymous

/* Povolení poštovních klientů, kteří používají zastaralé
autentizační příkazy, např. Microsoft Outlook. */
broken_sasl_auth_clients = yes

/* Posílání emailových zpráv přes tento poštovní server
povolíme pouze uživatelům důvěryhodných sítí a uživatelům,
kteří se autentizovali, ostatní zprávy jsou odmítnuty */
smtpd_recipient_restrictions =
                            permit_sasl_authenticated,
                            permit_mynetworks,
                            reject_unauth_destination
```

Protože emailové zprávy jsou mezi klienty a servery přenášeny přes veřejnou síť internet, je vhodné zajistit komunikaci proti odposlechu a narušení komunikace pomocí šifrování. jak popisuje [12]. Toho dosáhneme při použití TLS[43] , jehož podrobný popis je možno nalézt v [16]. Aby bylo možné ověřit totožnost poštovního serveru, je potřeba stejně jako pro www server Apache vygenerovat certifikát a zapnout podporu pro TLS. Pro vygenerování certifikátu spustíme v adresáři /etc/ssl/misc následující skripty(v této části se předpokládá, že byl vyplněn soubor openssl.cnf, jak bylo popsáno v kap. 2.2.7). Vygenerované certifikáty poté nakopírujeme do

43 **TLS** (Transport Security Layer) – Umožňuje aplikacím zabezpečenou komunikaci za použití kryptografie. Obecně bývá zaručena pouze totožnost serveru, klient se neověřuje. Více v [16].

složky, kde je bude očekávat Postfix :

```
/* Generován9 a podepsání certifikátu a vlastní
certifikikační autoritou */
# ./CA.pl -newreq-nodes
# ./CA.pl -newca
# ./CA.pl -sign

// Kopírování certifikátů
# cp newcert.pem /etc/postfix/ssl/
# cp newkey.pem /etc/postfix/ssl/
# cp demoCA/cacert.pem /etc/postfix/ssl
```

Dále v hlavním konfiguračním souboru main.cf nastavíme podporu TLS a cestu k certifikátům.

```
// Zapnutí TLS
smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_use_tls = yes
/* Klientům nebo serverům, kteří nepodporují TLS není
umožňeno komunikovat s Postfixem */
smtpd_tls_auth_only = yes

// Nastavení cest k certifikátům
smtpd_tls_key_file = /etc/postfix/ssl/newkey.pem
smtpd_tls_cert_file = /etc/postfix/ssl/newcert.pem
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
```

2.3.7 Instalace antispamového a antivirového filtru

Antivirový program Clamav Antivirus

Clamav Antivirus je Opensource antivirový program šířený pod licencí GPL. Je navržený hlavně pro scanování emailů na poštovních serverech. Obsahuje několik nástrojů pro zajištění maximálního zabezpečení přijímaných emailů. Například pro zajištění stále aktuální virové databáze slouží nástroj **Freshclam,** o jehož pravidelné spouštění se stará systémový nástroj Cron.

Antispamový filtr Spamassassin

Spam neboli nevyžádaná pošta je již delší dobu velkým problém celosvětové sítě internet. Některé firmy odhadují, že spam tvoří až 90% veškeré emailové komunikace. Spamassassin je Opensource antispamový filtr, který vyvíjí sdružení Apache. Spamassassin přiděluje jednotlivým příchozím zprávám trestné body, které se nakonec sečítají a pokud přesáhnou určitou hranici, je zpráva považována za spam. To, podle jakých znaků Spamassassin uděluje body, může administrátor do velké míry ovlivnit. Rozeznávají se například otazníky na konci předmětu, to že mail obsahuje pouze obrázek atd. Filtr dále umí spolupracovat se servery, které poskytují databázi

známých spamerů. Tato databáze je pravidelně aktualizována, což výrazně zlepšuje šanci na zachycení nevyžádané pošty. Filtr lze také nakonfigurovat tak, aby se sám „učil“. Obecně lze říci, že správně nakonfigurovaný spamassassin zachytí až 90% nevyžádané pošty.

Amavisd-new

Oba dva výše zmíněné programy, tedy antivirus Clamav a antispamový filtr Spamassassin, mohou pracovat samy a nebo být mohou spouštěny v případě potřeby. Amavisd-new je rozhraní mezi MTA,v našem případě tedy Postfixem, a programy, které zajišťují kontrolu pošty. Celý Amavisd-new je napsaný v jazyce Perl. Tento program tedy řídí činnost obou dvou filtrů obsahu. Postfix nejdříve přijatou zprávu předá do Amavisd-new, který zajistí spuštění příslušných kontrol a po vyhodnocení je zpráva dále předána Postfixu pro doručení, zahozena nebo upravena. To, co se s příchozí zprávou stane, samozřejmě záleží na výsledcích jednotlivých testů a na tom, jak je Amavisd-new nakonfigurován.

2.3.8 Instalace Amavisd-new, Clamav a Spamassassin

Všechny tři programy jsou součástí Portage stromu a můžou být pomocí emerge nainstalovány do systému, jak uvádí [6].

```
# emerge amavisd-new
# emerge spamassassin
# emerge clamav
```

Po zkompilování a nainstalování softwaru je potřeba zajistit, aby postfix předával příchozí zprávy do Amavisd-new před doručením do schránky uživatele. To zajistíme parametrem content_filter, který přidáme do konfiguračního souboru main.cf. Samotný Amavisd-new poté nakonfigurujeme tak aby naslouchal na portu 10024.

```
# content_filter = smtp-amavis:[127.0.0.1]:10024
```

Dále je potřeba otevřít další port, na kterém bude Amavisd-new předávat již zkontrolované zprávy zpět poštovnímu serveru Postfix. Do konfiguračního souboru postfixu master.cf přidáme podle [6] tyto řádky.

```
127.0.0.1:10025   inet   n   -   y   -   -   smtpd
   -o content_filter=
   -o local_recipient_maps=
   -o relay_recipient_maps=
   -o smtpd_restriction_classes=
   -o smtpd_helo_restrictions=
   -o smtpd_sender_restrictions=
   -o smtpd_recipient_restrictions=permit_mynetworks,reject
   -o mynetworks=127.0.0.0/8
   -o strict_rfc821_envelopes=yes
```

Závěr

Cílem této diplomové práce bylo prostudovat možnosti řešení internetových hostingových služeb, například poskytování prostoru pro www prezentace, poskytování emailových schránek, databázové služby a další doplňky k těmto službám, a poté se na základě získaných poznatků pokusit realizovat server, na kterém budou tyto služby provozovány. Podrobná specifikace služeb, které má server poskytovat, byla provedena firmou Maxprogres s. r. o., s jejíž spoluprací vznikl celý projekt. Použití volně šiřitelného software bylo další podmínkou, kterou firma Maxprogres s. r. o. stanovila. Dále bylo specifikováno, že běh serveru by měl co možná nejvíce zautomatizován a správa uživatelů realizována jednoduchým způsobem.

Po prostudování problematiky byly vybrány jednotlivé programy vykonávající konkrétní služby. Vlastní realizace serveru byla provedena celkem třikrát. První proběhla na mém soukromém pc a doméně kolonoh.net. Použitá linuxová distribuce Fedora Core 7 však nevyhovovala požadavkům firmy, která dále přidala požadavek na použití distribuce Gentoo, která je použita na většině serverů této firmy. Druhá realizace tedy proběhla na stejném pc a doméně s operačním systémem Gentoo. Server byl po odladění a nastavení služeb několik týdnu testován a poté po bezproblémovém chodu byla provedena třetí realizace na pc poskytnutém firmou Maxprogres.

Na serveru bylo aktivováno několik fiktivních testovacích účtů, pomocí kterých byly testovány jednotlivé komponenty systému v průběhu několika měsíců. Při testování serveru jsem začal pracovat na www rozhraní, kterým je možné ovládat jednotlivé služby a uživatelské účty na serveru. Realizace tohoto rozhraní je však již nad rámec této diplomové práce a uvádím ji zde pouze proto, aby si čtenář mohl udělat komplexní představu o celém projektu.

V březnu 2008 byl server předán firmě Maxprogres pro jejich potřeby a umístěn do jejich web-housingového centra.

Seznam použitých zkratek

ajax	Asynchronous JavaScript and XML
bios	Basic Input-Output System
des	Data Encryption Standard
DoS	Denial of Service
ftp	File Transfer Protocol
grub	Grand Unified Bootloader
http	Hypertext Transfer Protocol
https	Hypertext Transfer Protocol over Secure Socket Layer
ide	Integrated Drive Electronics
imap	Internet Message Access Protocol
ip	Internet Protocol
ldap	Lightweight Directory Access Protocol
lilo	Linux Loader
lvm	Logical Volume Manager
mbr	Master Boot Record
md5	Message-Digest algorithm 5
os	Operační systém
pam	Pluggable Authentication Modules
php	Hypertext Preprocessor
pop3	Post Officce Protocol verze 3
raid	Redundant Array of Independent Disks
sata	Serial Advanced Technology Attachment
scsi	Small Computer System Interface
sha	Secure Hash algorithm
smtp	Simple Mail Transfer Protocol
sql	Structured Query Language
ssl	Secure Socket Layer
tls	Transport Security Layer
url	Unique Resource Locator
www	World Wide Web

Seznam použité literatury

[1] CASTAGLIA .: *ProFTPD module mod_quotatab_sql* [online] [cit. 2008-04-25].
URL: <http://www.castaglia.org/proftpd/modules/mod_quotatab_sql.html>

[2] CINGROŠ, M.: *LVM2 - dynamické vytváření diskových oddílů* [online] [cit. 2008-04-25].
URL: <http://www.abclinuxu.cz/clanky/system/lvm2-dynamicke-vytvareni-diskovych-oddilu>

[3] DENT, D.: *Postfix – kompletní průvodce* , Praha 2005 ISBN 80-247-1029-3

[4] FLICKENGER, R.: *Linux server na maximum*, Brno 2005 ISBN 80-251-0586-5

[5] GENTOO FOUNDATION .: *Gentoo Handbook* [online][cit. 2008-04-25].
URL: <http://www.gentoo.org/doc/en/handbook>

[6] GENTOO FOUNDATION .: *Virtual Mailhosting System with Postfix Guide* [online]
[cit. 2008-04-25]. URL: <http://www.gentoo.org/doc/en/virt-mail-howto.xml>

[7] HATCH, B., LEE J., KURTZ G.: *Hacking bez tajemství - Linux*, Brno 2003
ISBN 80-7226-896-4

[8] KABIR M. J.: *Apache server 2* , Brno 2004 ISBN 80-251-0319-6

[9] KOCMAN, J.: *Jak na démona Cron* [online][cit 2008-05-15].
URL: <http://interval.cz/clanky/jak-na-demona-cron/>

[10] KOLÍSEK, A.: *Cesta do hlubin kompilace jádra* [online][cit 2008-05-15]
URL: <http://www.abclinuxu.cz/clanky/navody/cesta-do-hlubin-kompilace-jadra-1>

[11] KRČMÁŘ, P.: *Na co se často ptáme - /etc/fstab* [online][cit 2008-05-15].
URL: <http://www.abclinuxu.cz/clanky/system/na-co-se-casto-ptame-etc-fstab>

[12] LOCKHART, A.:, *Bezpečnost sítí*, Brno 2005 ISBN 80-251-0805-8

[13] PETŘÍČEK, M.: *Stavíme firewall* [online][cit 2008-04-14].
URL: <http://www.root.cz/clanky/stavime-firewall-1/>

[14] POSTFIX ORG.: *Postfix Documentation* [online][cit 2008-04-14].
URL: <http://www.postfix.org/documentation.html>

[15] PROFTPD .: *Proftpd Documentation* [online][cit 2008-04-14].
URL: <http://proftpd.org/docs/>

[16] WIKIPEDIA .: Transport Layer Security [online][cit 2008-04-14].
URL: <http://cs.wikipedia.org/wiki/Transport_Layer_Security>

[17] WIKIPEDIA .: Master boot record [online][cit 2008-04-14].
URL: <http://cs.wikipedia.org/wiki/Master_boot_record>

[18] WIKIPEDIA .: RAID [online][cit 2008-04-14]. URL: <http://cs.wikipedia.org/wiki/RAID >

[19] YOSHINORI ,O.: GNU GRUB [online][cit 2008-04-14].
URL: < http://www.gnu.org/software/grub/>