VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

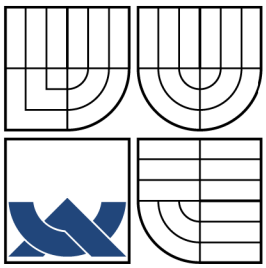
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMUNICATIONS

NÁVRH A REALIZACE MAIL-SERVERU S VYUŽITÍM MYSQL.

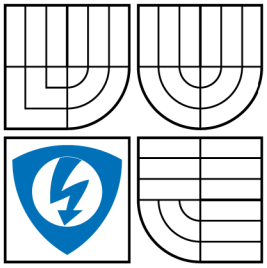
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE PETR SMAHEL
AUTHOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMUNICATIONS

NÁVRH A REALIZACE MAIL-SERVERU S VYUŽITÍM MYSQL.

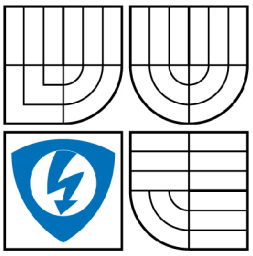
DESIGN AND IMPLEMENTATION OF MAIL-SERVER WITH MYSQL.

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PETR SMAHEL
AUTHOR

VEDOUCÍ PRÁCE — ING. MOJMÍR JELÍNEK
SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Petr Smahel ***ID:*** 78752
Ročník: 3 ***Akademický rok:*** 2008/2009

NÁZEV TÉMATU:

Návrh a realizace mail-serveru s využitím MySQL

POKYNY PRO VYPRACOVÁNÍ:

Realizace komletního řešení poštovního serveru pod operačním systémem Linux, využívajícího databázi MySQL. Testování výkonu a limitů daného řešení. Cílem je také srovnání výkonnosti a stability s jinými operačními systémy.

DOPORUČENÁ LITERATURA:

[1] NEMETH, E., SNYDER, G., HEIN T. Linux - Kompletní příručka administrátora. Computer Press, 2004. 880 s. ISBN: 80-722-6919-4.

[2] ŠŤASTNÝ, Petr. Typy tabulek v MySQL [online]. 27.03.2007 [cit. 2008-04-08]. Dostupný z WWW: http://www.pweb.cz/a/14/typy-tabulek-v-mysql.html.

Termín zadání: 9.2.2009 ***Termín odevzdání:*** 2.6.2009

Vedoucí práce: Ing. Mojmír Jelínek

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

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. **Pan/paní**

   Jméno a příjmení: Petr Smahel

   Bytem: č. p. 471, 756 43 Kelč

   Narozen/a (datum a místo): 23.3.1986, Valašské Meziříčí

(dále jen "autor")

a

2. **Vysoké učení technické v Brně**

   Fakulta elektrotechniky a komunikačních technologií

   se sídlem Údolní 244/53, 60200 Brno 2

   jejímž jménem jedná na základě písemného pověření děkanem fakulty:

   prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

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

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

(dále jen VŠKP nebo dílo)

Název VŠKP: Návrh a realizace mail-serveru s využitím MySQL

Vedoucí/školitel VŠKP: Ing. Mojmír Jelínek

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

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

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

Článek 2

Udělení licenčního oprávnění

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

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

Článek 3
Závěrečná ustanovení

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

V Brně dne: .....

....

Nabyvatel Autor

ABSTRAKT

Práce se zabývá návrhem a realizací kompletního poštovního serveru na platformě GNU/Linux, sestaveného z kvalitního opensource softwaru a využívajícího databazový systém MySQL. První část práce pojednává o obecné problematice elektronické pošty v prostředí Internetu, dále pak o charakteristice operačního systému GNU/Linux a relačních databázových systémů. V druhé části práce se lze seznámit s konkrétními softwarovými produkty využívanými pro realizaci poštovních serverů, instalace a konfigurace těchto produktů je popsána v závěrečné třetí části.

KLÍČOVÁ SLOVA

Linux, MySQL, Postfix, e-mail, server

ABSTRACT

The thesis involves the design and implementation of a complete mail server using the GNU/Linux platfrom built on quality opensource software, using MySQL database. The first part deals with general problem topics concerning internet electronic mail and also the characteristics of GNU/Linux and related database systems. The second part deals with getting acquainted with concrete mail server implementations and products, while the third part deals with the installation and configuration of these products.

KEYWORDS

Linux, MySQL, Postfix, e-mail, server

SMAHEL P. *Návrh a realizace mail-serveru s využitím MySQL.. Brno: Vysoké Učení Technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2009. 63 s., 15 s. příloh. Bakalářská práce. Vedoucí práce byl Ing. Mojmír Jelínek.*

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh a realizace mail-serveru s využitím MySQL.“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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)

PODĚKOVÁNÍ

Chtěl bych poděkovat všem, kteří přispěli ke vzniku této práce, zejména pak vedoucímu bakalářské práce Ing. Mojmíru Jelínkovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování úkolu.

V Brně dne ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Elektronická pošta je jednou z nejstarších a nejoblíbenějších služeb v oblasti Internetu. Lidé po celém světě si ji oblíbili, protože se pohodlně používá, je velmi rychlá a levná. Vzdálenosti mezi lidmi se díky ní zkrátily a dopis, který by dříve k adresátovi putoval dny i týdny, je dnes ve formě e-mailu doručen během několika vteřin. Podobným způsobem, jakým je klasický dopis předáván mezi poštovními stanicemi, putuje i e-mailová zpráva sítí z jednoho systému do druhého, aby nakonec byla doručena do schránky adresáta. Tyto systémy jsou obvykle označovány jako poštovní servery, nebo také mail-servery.

Mechanizmy, pomocí kterých mezi sebou poštovní servery komunikují a předavájí si jednotlivé e-mailové zprávy, jsou přesně definovány standarty a protokoly. Této problematice se věnuje první část práce, která je pojata především jako teoretický úvod do problematiky elektronické pošty. Popisuje architekturu elektronické pošty, její hlavní součásti a pravidla pro kumunikaci mezi nimi.

V dnešní době velká část poštovních serverů používá jako základ operační systém GNU/Linux, jehož výhody, nevýhody a charakteristické vlastnosti jsou zhodnoceny v druhé kapitole.

Třetí kapitola je věnována relačním databázím, zejména pak databázovému systému MySQL, jehož škála použití je velmi široká a zasahuje i do oblasti elektronické pošty.

Pro realizaci poštovního serveru existuje množství různých softwarových produktů, lišících se kvalitou i cenou. Druhá část práce se zabývá návrhem softwarového vybavení poštovního serveru. V jednotlivých kapitolách jsou popsány zvolené softwarové produkty, jejich stručná charakteristika a shrnutí nejdůležitějších vlastností.

Třetí část je věnováná praktické realizaci poštovního serveru, zejména pak instalaci a konfiguraci vybraného softwaru a testovaní jednotlivých služeb poskytovaných mail-serverem.

1 TEORETICKÉ ŘEŠENÍ

Elektronická pošta jako služba pracuje nad systémem, který se skládá z několika částí a je řízen přesně definovanými pravidly. Tato část práce popisuje součásti elektronické pošty a pravidla pro jejich komunikaci, dále se věnuje operačnímu systému GNU/Linux a relačním databázím, zejména pak databázovému systému MySQL.

1.1 Elektronická pošta

Historie internetové elektronické pošty (e-mailu) sahá až do 70. let minuleho století, kdy docházelo k odesílání prvních zpráv přes síť Arpanet[1]. Od té doby je e-mail nejpoužívanější aplikací na intenetu. Kdysi bylo doručovaní elektronické pošty relativně jednoduché a obvykle sestávalo z přesunování malých poštovních souborů z jednoho velkého hostitele na jiného velkého hostitele, který sloužil mnoha uživatelům. Tyto soubory byly psány výhradně anglicky, tedy bez háčků a čárek.

S postupem času a s rostoucí oblibou elektronické pošty pak došlo k jejímu vylepšením, a to hned v několika směrech. Nově byla zavedena podpora i jiných jazyků, resp. znakových sad, byl ujednocen způsob přibalování příloh a posléze se rozšířil i repertoár formátů, které může mít samotný obsah zprávy. Díky těmto změnám se pak z elektronické pošty stala mnohem univerzálnější služba, tedy platforma pro poskytování dalších specifických služeb.

1.1.1 Achitektura elektronické pošty

Elektonická pošta má, stejně jako většina internetových služeb, architekturu klient-server. Počítá tedy s dvoučlenným dělením práce. Jedna část se zábývá vlastním doručováním zpráv, druhá zajišťuje potřebnou součinost s uživatelem. Funkci serveru plní agent MTA, například programy Sendmail, Postfix, Exim, Qmail. Funkci klienta agent MUA, příkladem agentů MUA mohou být programy Thunderbird, Outlook Express, Pine, Kmail.

Samotný agent MTA bývá často doplněn o další specializovaný software, jako například virový a spamový filtr, POP3/IMAP server, agenta MDA, software pro třídění pošty, software pro stahování pošty z jiných schránek atd. a společně tvoří poštovní server.

[1] Advanced Research Projects Agency Network byla počítačová síť spuštěná v roce 1969, je považována za předchůdce dnešního Internetu. Odpojena byla v roce 1990.

Agenti elektronické pošty

MUA – Mail User Agent (poštovní uživatelský agent) je klientský software pro elektronickou poštu určený pro vytváření, odesílání a přijímání zpráv elektronické pošty. Odesílá zprávy prostřednictvím MTA. Zprávy načítá z úložiště pošty buď přímo, nebo skrze server POP3/IMAP.

MTA – Mail Transfer Agent (poštovní přenosový agent) je server, který přijímá a doručuje elektronickou poštu. Určuje směrování zprávy a možné přepsání adresy. Lokálně doručované zprávy jsou předány MDA pro jejich konečné doručení.

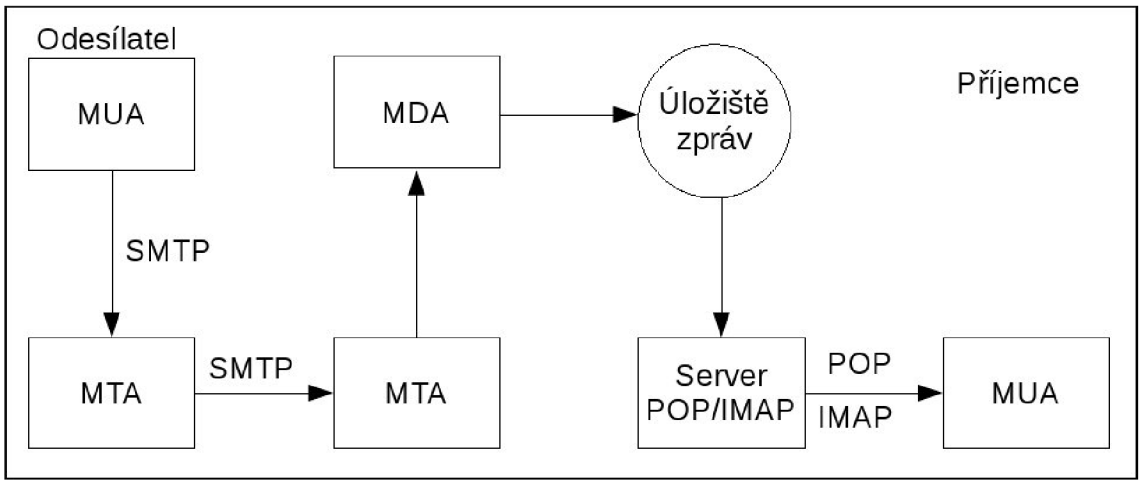
MDA – Mail Delivery Agent (agent doručení zprávy) je program, jenž provádí konečné doručení zpráv pro místní příjemce systému. MDA zprávy při doručování často filtruje nebo zařazuje do kategorií. MDA může také určovat, zda má být zpráva předána na jinou e-mailovou adresu.

Obrázek 1.1 ukazuje jednotlivé komponenty, které se učastní jednoduchého vyslání e-mailu od odesílatele k přijemci. Agenti MTA mají na starosti veškerou práci s přesunem z jednoho systému do druhého. Když obdrží požadavek na příjem zprávy, stanoví agent MTA, zda má danou zprávu přijmout nebo nikoli. MTA obvykle přijímá zprávy nejen pro své vlastní lokální uživatele, ale také pro jiné systémy, jimž umí zprávu předat, nebo zprávy od uživatelů, systémů či sítí, které mohou předávat poštu do jiných cílů. Jakmile daný MTA zprávu přijme, může ji buď poslat nějakému uživateli na svém systému, nebo ji předat dál jinému agentovi MTA. Jakmile zpráva dorazí na konečný MTA a je určena uživateli na tomto systému, daný MTA ji předá agentovi MDA, například Procmail, Maildrop, ten poté zajistí její konečné doručení. MDA může zprávu uložit jako prostý soubor, nebo ji může předat speciální databázi. Obě tyto možnosti lze spojit termínem Message Store (úložiště zpráv).

Zpráva zůstane umístěna v úložišti do té doby, dokud není příslušný uživatel připraven k jejímu vyzvednutí. K převzetí zprávy využívá příjemce agenta MUA. Ten kontaktuje server POP3/IMAP, jenž poskytuje přístup k úložišti zpráv. Tento server je oddělen od MTA, jenž správu dodal, a je vytvořem specificky k zajištění přístupu pro přebírání zpráv. Po úspěšném ověření žadatele, může server zprávu tohoto žadatele odeslat jeho agentovi MUA.

RFC

Pravidla pro komunikaci mezi popisovánými komponentami e-mailového systému definují standardy a protokoly. Dokumenty standardů spravuje skupina Internet



Obr. 1.1: Cesta e-mailu od odesílatele k příjemci.

Engineering Task Force (IETF) a publikuje je jako Request For Comments (RFC – žádost o komentář). Jsou to číslované dokumenty vysvětlující určitou technologii nebo protokol [7].

DNS a směrování pošty

Systém DNS (Domain Name System)[2] je nesmírně rozsáhlá distribuovaná databáze, jejímž hlavním úkolem je mapování názvů hostitelů na adresy IP. Celý systém je tvořen hirerarchicky a data jsou distribuována, což znamená, že každý server aktualizuje své vlastní informace a aktualizace jsou dostupné takřtka okamžitě. Hirerarchické pojmenovaní zabraňuje konfliktům v názvech hostitelů a poskytuje aktuální systém názvů domén, tak jak je znám dnes. Obrázek 1.2 ukazuje příklad doručení e-mailu, kdy zdrojový MTA využíva pro zjištění IP adresy cílového MTA DNS server.

Data se skládají z různých druhů záznamů, které se nazývají Resource Records (záznamy prostředků). Existuje několik typů záznamů prostředků, každý poskytuje různý druh informace.

Přehled nejdůležitějších DNS záznamů

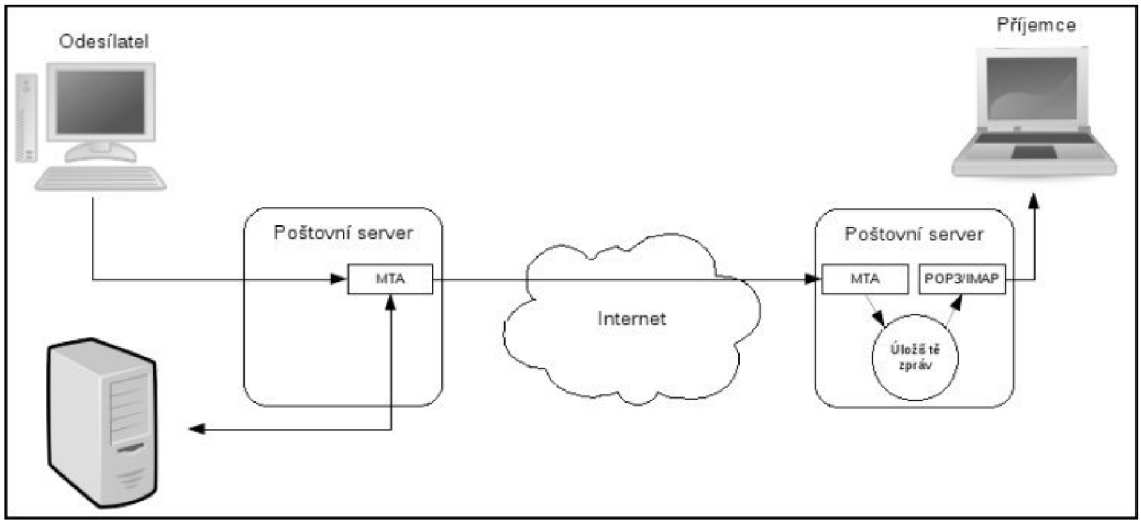
A – provádí mapování názvů domén na IP adresy. A záznamy obsahují název hostitele a jeho IP adresu. Například záznam `utko.cz A 147.229.199.13` převede hostitele utko.cz na adresu 147.229.199.13.

CNAME – názvem hostitele může být i alias, který ukazuje na jiné názvy hostitelů namísto IP adres. To může být užitečné při směrování požadavků na

[2]DNS systém byl poprvé definován v roce 1983 v dokumentech RFC 882 a RFC 883, aktualizován byl v roce 1987 v dokumentech RFC 1034 a RFC 1035.

služby. Záznam CNAME poskytuje skutečný nebo kanonický název, na který alias názvu hostitele ukazuje. Například CNAME záznam `www.utko.cz CNAME server1.utko.cz` nám říká, že název hostitele www.utko.cz je aliasem názvu server1.utko.cz.

MX – záznamy MX (mail exchange), udávají názvy poštovních serverů, jež se starají o veškerou poštu pro danou doménu. Jelikož může mít jedna doména více poštovních serverů, obsahují záznamy MX hodnotu pro stanovení pořadí priority. Server s vyšší prioritou (nižší hodnota) bude při výběru preferován. Například z MX záznamu `utko.cz MX 10 mail.utko.cz` je patrné, že poštovní server pro doménu utko.cz nese název mail.utko.cz a má prioritu 10.



Obr. 1.2: Doručení e-mailu v síti.

Obálka zprávy

Je potřeba rozlišovat mezi adresou obálky zprávy a adresou záhlaví zprávy. Doručení zprávy řídí adresa obálky. Z pohledu MTA je záhlaví zprávy součástí obsahu e-mailu. Obálku zprávy vytváří MTA při odesílání zprávy dále. Obálka musí obsahovat minimálně tyto dva údaje: adresu odesílatele (MAIL FROM) a adresu příjemce (RCPT TO). Je důležité si uvědomit, že adresa příjemce uvedená na SMTP obálce nemusí být shodná s adresátem vlastní zprávy. Důvodů k tomu je hned několik – například ten, kdy jedna a tatáž zpráva má kromě svého hlavního příjemce (vyjádřeného v záhlaví zprávy v položce To:) také jednoho či několik příjemců běžných či slepých kopií. Když je pak zpráva doručována těmto příjemcům kopií, je na obálce nadepsána jejich adresa, zatímco v položce To: v záhlaví zprávy zůstává nadále adresa hlavního adresáta.

Formát e-mailové adresy

Formát e-mailových adres je podrobně popsán v dokumentu RFC 5322. E-mailová adresa se skládá ze tří základních částí:

místní část – jméno uživatele nebo alias na jinou adresu, označována jako levá strana

oddělovač – znak @, tzv. zavináč

název domény – označována jako pravá strana

Příkladem e-mailové adresy je: *pavelnovak@utko.cz*.

Formát e-mailové zprávy

RFC 5322 rovněž udává formát e-mailových zpráv. Zpráva se skládá ze dvou částí: záhlaví a těla. Záhlaví obsahuje pole s názvy, jako např. To (komu), From (od), Subject (předmět) následovanými dvojtečkou (:), za ní je obsah daného pole. Jedno pole může zabírat více řádků, pokračující řádky začínají bílým znakem (tab nebo mezera). Jedinými povinnými poli jsou Date: a From:. Záhlaví je od těla odděleno prázdným řádkem. Tělo obsahuje samotný obsah zprávy, má volný formát, ale mělo by obsahovat pouze ASCII znaky. Následuje příklad formátu e-mailové zprávy převzatý z [8].

```
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>

This is a~message just to say hello.
So, "Hello".
```

Rozšíření standardem MIME (Multipurpose Internet Mail Extensions) přineslo možnost posílání zpráv s diakritikou, obrázků, zvuků atd. MIME, definuje jak převést tato data na ASCI znaky. Typ přenášených dat určuje položka MIME Content-Type (zkráceně MIME-type). Jedná se o znakový řetězec obsahující tři složky:

typ – definuje o jaký typ souboru se jedná (text, zvuk, obrázek,...)

podtyp – definuje formát souboru

doplňkové informace – mohou obsahovat např. parametr udávající hodnotu

Příklad Content-Type: *image/jpeg; parametr1=hodnota;*

1.1.2 Hlavní protokoly elektronické pošty

Pro komunikaci mezi jednotlivými komponentami elektronické pošty se používají tři hlavní protokoly. K odesílání zpráv se používá protokol SMTP, zatímco k jejich příjmu slouží protokoly POP3 nebo IMAP.

Protokol SMTP

Simple Mail Transport Protocol (SMTP - jednoduchý protokol přenášení pošty)[3], je určen k předávání pošty od klienta k serveru a k přenosu pošty mezi servery. Při předávání zprávy vystupuje odesílající server jako klient vzhledem k přijímajícímu serveru. Komunikace standardně probíhá na TCP portu 25.

Protokol SMTP chápe přenášená data jako textová, členěná na jednotlivé řádky (pomocí znaků CR a LF), a tvořená pouze znaky z původní 128prvkové abecedy ASCII. Jinými slovy: SMTP předpokládá pouze přenos znaků, kódovaných do sedmi bitů. Pokud se takovéto sedmibitové znaky přenáší kanálem, který je uzpůsoben přenosu osmibitových znaků resp. bytů (což je mj. příklad protokolu TCP) pak standard SMTP definuje, že jeho sedmibitové znaky mají být vkládány do osmic bitů tak, aby byly zarovnány doprava a zleva doplněny nulovým bitem [11].

Pomyslná obálka protokolu SMTP se svými údaji je přitom přenášena také ve formě textu, a to na začátku přenosu. Celá komunikace mezi příjemcem a odesilatelem (po navázání spojení na úrovni protokolu TCP) má formu dialogu, v rámci kterého se obě strany nejprve informují o své obecné připravenosti přijímat poštu, pak si předají údaje o odesilateli a příjemci (resp. další údaje z pomyslné obálky), a poté pak i samotný obsah zprávy [11].

Dokument RFC 1869 umožňuje rozšíření základního SMTP o další funkce. Rozšířený protokol je označován jako ESMTP. Použitím příkazu EHLO namísto HELO klient oznamuje, že je schopen používat ESMTP. Pokud server také podporuje rozšíření, odpoví seznamem funkcí, jenž podporuje.

Protokol POP3

Post Office Protokol version 3 (POP3 - poštovní protokol verze 3)[4] je protokol, který se používá pro stahování emailových zpráv ze vzdáleného serveru do MUA agenta uživatele.

Protokol POP3 není nijak zabezpečen, avšak podruje několik autentizačních metod ověřování na různých úrovních ochrany před neoprávněným přístupem k cizí

[3]SMTP byl původně definován v RFC 821, později byl rozšířen v RFC 2821 a v RFC 5321.
[4]POP3 protokol byl standardizován v roce 1996 v RFC 1939.

poštovní schránce. Jedna taková metoda, APOP (kterou základní specifikace definuje jako volitelný příkaz), užívá MD5 hash funkci pro zabezpečený přenos hesla od klienta na server. Klienti mohou také kódovat celou POP3 komunikaci použitím SSL nebo modernějšího TLS.

Komunikace probíhá na principu výměny zpráv mezi klientem a serverem. V základní implementaci POP3 mají příkazy 3 nebo 4 znaky. Příkazy nerozlišují velká a malá písmena. Za příkazem můžou následovat další argumenty, oddělené mezerami. Řádky jsou oddělovány pomocí znaků CR a LF. Každá odpověď od serveru musí začínat indikací stavu operace - buď +OK, nebo -ERR, za kterou může následovat textový řetězec s popsaným důvodem stavu. Protokol POP3 má pro své účely vyhrazen TCP port 110.

Protokol IMAP

Internet Mail Aplication Protokol (IMAP - protokol aplikace internetové pošty)[5] je protokol pro přístup k e-mailovým schránkám.

Na rozdíl od POP3 se standardně nepřenáší veškerá pošta k uživateli, ale stahuje se jen to, o co má uživatel aktuálně zájem. Rozdíly zahrnují podporu pro práci více připojených klientů zároveň, uchovávání stavů zpráv na serveru, podporu více složek včetně zanoření, prohledávání zpráv na straně serveru nebo například zobrazování obsahů zpráv bez příloh a tím šetřit linku. Výhodou protokolu IMAP je možnost pracovat s poštou na různých místech (různých počítačích nebo přes webové rozhraní) a stále mít k dispozici stejné zprávy a stejnou strukturu složek. IMAP4 určuje explicitní mechanismus, podle kterého může být rozšířen. Nevýhodou je značná složitost protokolu, což vede k častějším chybám.

IMAP4 má zabudovanou podporu šifrovaného přihlášení, ale umožňuje i přenos nezakódovaného hesla, neboť použití šifrovacího mechanismu musí být odsouhlaseno serverem i klientem a v některých případech je nutné použít nezakódované heslo. Komunikaci IMAP4 lze také zakódovat pomocí SSL. Buď se komunikuje přes SSL tunel na portu 993, nebo se při komunikaci přes IMAP4 použije STARTTLS (Transport layer security). Protokol IMAP standardně používá port 143 protokolu TCP.

1.2 Operační systém GNU/Linux

GNU/Linux patří do rodiny unixových operačních systémů. Mezi další unixové oprační systémy patří například BSD, Mac OS, Solaris apod.

[5] V současné době se používá verze IMAP4 (IMAP version 4 revision 1 - IMAP4rev1) definována v RFC 3501.

1.2.1 Historie

Projekt GNU[6] založil v roce 1983 Richard Stallman[7]. Jeho cílem bylo vytvořit nový operační systém unixového typu, který by byl složen jen ze svobodného software. Za tímto účelem sepsal Stallman novou licenci GNU GPL (GNU General Public License), pod kterou jsou šířeny všechny části systému GNU.

Během necelých deseti let se GNU stal zcela použitelným systémem kompatibilním s komerčními unixy. Chybělo jen jádro, které by zajistilo samotný běh systému a komunikaci s hardware. Proto byl v roce 1990 zahájen vývoj jádra Hurd.

V roce 1991 začal finský student Linus Torvalds[8] pracovat na vývoji vlastního unixového jádra, odvozeného od unixového operačního systému Minix. Jeho jádro, jež nakonec dostalo jméno Linux, si okamžitě našlo řadu příznivců, kteří se začali na jeho vývoji aktivně podílet. Linus se později rozhodl zdrojové kódy uvolnit pod svobodnou licencí GNU GPL.

Jelikož byl vývoj Linuxu mnohem úspěšnější než vývoj Hurdu, začal se operační systém GNU používat společně s jádrem Linux. Tím vznikl výsledný produkt s názvem GNU/Linux.

Spojením GNU, Linuxu a dalších projektů vznikají tzv. distribuce, jež jsou kompilací jednotlivých částí, a tvoří tak komplexní operační systém.

1.2.2 Charakteristika

GNU/Linux je především svobodný, sofistikovaný a univerzální systém. Může být nasazen do embedded zařízení, na osobní počítače či servery, ale i na sálové počítače. Byl portován na širokou paletu architektur, od běžné x86 (osobní počítače 386 a vyšší), přes PowerPC (MacIntosh), Sparc (Sun Microsystems), Motorola 68k až třeba po MIPS. Podporuje multitasking (běh několika programů současně) a je víceuživatelský (několik uživatelů může současně pracovat na jednom fyzickém systému). Stejně dobře pracuje jak v textovém, tak i v grafickém režimu. Používá moderní žurnálovací souborové systémy, jako například ext3, ReiserFS, XFS.

GNU/Linux je využíván hlavně v serverových instalacích bez grafického uživatelského rozhraní, pro svou stabilitu, snadnou vzdálenou zprávu a v neposlední řadě pro nízké pořizovací náklady. GNU/Linux si ale pomalu hledá své místo i na počítačích bežných uživatelů, kde jeho podíl v poslední době prudce roste.

[6]Rekurzivní zkratka vytvořena z prvních písmen anglických slov GNU's Not Unix.

[7]Richard Matthew Stallman (* 16. března 1953, Manhattan, New York), známý též pod iniciálami RMS, je zakladatel hnutí svobodného softwaru, projektu GNU a v říjnu 1985 také Free Software Foundation.

[8]Linus Benedict Torvalds (* 28. prosince 1969) je finský programátor, známý především zahájením vývoje jádra operačního systému Linux a pozdější koordinací projektu.

Pro správu software využívá většina distribucí tvz. balíčkovací systémy. Ty zajišťují nejen instalaci a odinstalaci balíčků, ale řeší i závislosti mezi jednotlivými balíčky. Balíčky se obvykle nachází v tzv. repozitářích (zdrojích softwaru), odkud jej balíčkovací systémy stahují.

Jedním z typických rysů Unixových operačních systémů, včetně GNU/Linux je jejich adresářová struktura, kterou definuje standard FHS (Filesystem Hierarchy Standard) a dává tak možnost „předvídat“, kde se který soubor či adresář nachází[9].

Struktura adresářů v Linuxu

`/` – kořen souborového systému, začátek stromové struktury

`/bin` – základní spustitelné soubory, dostupné všem uživatelům

`/boot` – statické soubory zavaděče (boot loader)

`/dev` – soubory zařízení

`/etc` – globální konfigurační soubory systému

`/home` – domovské adresáře uživatelů

`/lib` – základní sdílené knihovny a moduly jádra

`/lost+found` – ztracené a opravené soubory po chybách souborového systému

`/mnt` – adresář pro dočasně připojované souborové systémy.

`/opt` – přídavné softwarové balíky

`/proc` – soubory nastavení a stavu systému a jednotlivých procesů

`/root` – domovský adresář superuživatele (root)

`/sbin` – systémové privilegované spustitelné soubory

`/sys` – virtuální adresář

`/tmp` – adresář pro odkládací a pomocné soubory

`/usr` – sekundární hierarchie adresářů konfigurační

`/var` – data podléhající změnám

[9] U jednotlivých distribucí se mouhou v adresářové struktuře objevovat různé odchylky.

1.2.3 Distribuce

Jak již bylo řečeno, GNU/Linux je šířen v distribucích. Distribuce je samotný operační systém GNU/Linux a velké množství aplikací. Existuje řada firem, organizací, skupin i jedinců, kteří sestavují a nabízejí své distribuce. Distribucí jsou v dnešní době stovky. Přehled některých z nich nejlépe ilustruje, jak široké jsou možnosti GNU/Linux.

Přehled vybraných linuxových distribucí

Debian – je svobodná nekomerční distribuce výhradně ze svobodného softwaru, která je navíc plně vyvíjena komunitou. Debian je znám svým specifickým vývojovým cyklem, kdy jsou k dispozici tři větve: stable, testing a unstable. Všechny softwarové balíčky, které se do distribuce dostanou, postupně prochází testy až do stabilní verze. Díky tomu jsou zde všechny DEB balíčky perfektně odzkoušené, na druhou stranu nejsou nejaktuálnější. Velkou výhodou Debianu je také jeho balíčkovací systém apt, který řídí veškeré instalace a řeší veškeré její problémy a potřebné kroky provádí za uživatele.

SuSE – lze dnes získat ve dvou podobách. Jako OpenSuSE, což je komunitou vyvíjená distribuce a jako SuSE Enterprise Linux, která představuje komerční variantu. Na obou se podílí firma Novell v rámci jejích linuxových aktivit. SuSE obsahuje instalační a administrační program YaST2, který hlídá online updaty, konfiguruje síť a firewall, administruje uživatele apod. SuSE je známé svou detekční schopností běžných winmodemů, notebooků a obecně dobrou podporou ovladačů. Je dodáván se spoustou softwaru a obsahuje tedy vše pro desktop a může být též využíván v podnikové sféře.

Gentoo – je source-based distribuce, tj. celý systém se kompiluje úplně od základu. Obsahuje však balíčkovací systém Portage, který veškerou kompilaci zjednodušuje. Gentoo umožňuje uživateli vytvořit si kompletní systém ze zdrojových kódů s optimalizacemi podle svých představ. Veškeré nastavení si musí uživatel provést sám, proto je tato distribuce vhodná pro pokročilé uživatele. Distribuce je velmi čistá, aplikace kompilované na míru jsou rychlé a obsahuje pouze to, co opravdu uživatel opravdu chce.

Knoppix – je bootovatelná live distribuce obsahující široký výběr software, automatickou detekci hardware, podporu pro mnoho grafických karet, zvukových karet a dalších periferií. Nevyžaduje instalaci na disk, všechny soubory vytvořené uživatelem se ukládají v operační paměti. Díky své univerzálnosti je

Knoppix použitelný jako produkční systém na desktopu, jako systém pro výuku, záchranný systém a podobně.

Ubuntu – je poměrně mladá distribuce vyvíjená komunitou za podpory firmy Canonical. Ubuntu je ovšem postaveno na velmi spolehlivém základě již mnoho let vyvíjené distribuce Debian GNU/Linux. Zatímco Debian se snaží být distribucí značně univerzální, jasným cílem Ubuntu je přiblížit Linux uživatelům pro použití na osobních počítačích jako tzv. desktopové prostředí. Grafické prostředí Ubuntu je založeno na Gnome, někteří lidé ale více preferují grafické prostředí KDE. Proto vznikla varianta Ubuntu zvaná Kubuntu. Odlehčená verze s Xfce neboli Xubuntu je vhodná např. pro starší počítače.

1.3 Relační databáze

Databáze je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. Databáze však není jen prosté shromaždiště, úložiště dat, ale většinou slouží zároveň k jejich organizaci, třídění, prohledávání, seskupování a podobně. K typické dnešní databázi patří rovněž to, že zároveň chce s daty pracovat více uživatelů.

Relační databáze se skládá z více objektů (tabulek). Tabulky vždy popisují určitou část reálného světa a stejně jako ve skutečném životě i předměty zachycované tabulkami mohou být spolu v nějakém vztahu. To znamená, že jedna tabulka může obsahovat identifikátor položky z jiné tabulky. Vzájemnému vztahu mezi tabulkami se říká relace.

Příklad relace mezi několika tabulkami

Tabulka `UČITEL` obsahuje jméno a příjmení učitele, dále může obsahovat jeho odborné schopnosti, věk a podobně. Kromě popisných dat je v tabulce také speciální sloupec s jednoznačným číselným identifikátorem učitele, který slouží jako klíč. Pomocí tohoto klíčového atributu je možné na konkrétního učitele odkazovat.

```
+----+-------+----------+-----+
| ID | Jméno | Příjmení | ... |
+----+-------+----------+-----+
|  1 |  Jan  |  Novák   |     |
| ...|
+----+-------+----------+-----+
```

Tab. 1.1: Tabulka `UČITEL`.

V tabulce TŘÍDA jsou uchovány informace o konktrétních třídách. Například stupeň třídy a její název (třeba A). Stejně jako v předchozí tabulce i zde má každá položka svůj jednoznačný identifikátor – klíč.

ID	Stupeň	Název	...
1	1	A~	
...			

Tab. 1.2: Tabulka TŘÍDA.

Obsahem tabulky UČÍ jsou dva sloupce, klíč učitele a klíč třídy. Záznam (2, 5) bude tedy znamenat, že učitel 2 učí třídu 5, z čehož je patrné, že učitel 2 je ve vztahu (recali) s třídou 5.

ID_učitele	ID_třídy
1	1
1	2
3	1
...	

Tab. 1.3: Tabulka UČÍ.

1.3.1 Základní pojmy

Databázový server – to je název softwaru, který zastřešuje a řídí jednotlivé databáze. Má na starosti takové věci, jako je autorizace uživatelů, poskytování a vracení dat, jejich organizace, sdílení po síti a správu.

Databázový klient – je název libovolného software, který s databázovým serverem komunikuje (tzn. buď mu předává nějaká data k uložení, nebo po něm data chce). Typicky se může jednat o řádkového klienta, o webovou stránku zobrazující data, o obchodní aplikaci nebo o cokoli podobného.

Databáze – termín pro označení kolekce tabulek. V praxi se databáze používají zejména pro logické oddělení vzájemně nesouvisejících tabulek a pro zjednodušení správy oprávnění.

Tabulka – je složena ze sloupců a řadků. Tabulka je základním stavebním kamenem pro budování celé databáze

Řádek – někdy též záznam nebo věta. Každý řádek obsahuje informace o jedné položce (například jméno, příjmení a telefon pro jednoho zaměstnance). Řádky lze do tabulky přidávat z klientské aplikace, z jiné tabulky nebo z externího zdroje (pak se tomu říká import).

Sloupec – prostor pro uložení jedné položky záznamů. V jednom sloupci v tabulce databáze musí být data stejného datového typu.

Primární klíč – je sloupec, který jednoznačně identifikuje záznam. To znamená, že tento sloupec musí být u každého záznamu jiný.

1.3.2 MySQL

MySQL je relační databázový systém typu DBMS (database managment system), který vychází z deklarativního programovacího jazyka SQL (Structured Query Language). MySQL byla vytvořen švédskou firmou MySQL AB a je šířena jako open source. MySQL je malý, rychlý a jednoduchý databázový systém. Databáze MySQL má některá omezení, která obsahují jiné databázové systémy, např. robusní Oracle. Právě díky tomu dosahuje vynikající rychlosti. MySQL nabízí několik typů tabulek. Každý typ má své výhody a nevýhody a hodí se pro různé situace. Jedním z hlavních rozdílů je podpora transakcí[10].

Systém MySQL disponuje některými specifickými vlastnostmi, díky kterým se stal rozšířenější než konkurenční databázové systémy, mezi něž patří například Oracle, PostgreSQL, SQLite, Firebird.

Vlastnosti MySQL

- Nejdůležitější vlastností databáze je její stabilita. MySQL je velmi stabilní. Každá nová verze je vždy vývojáři důkladně otestována.
- Další velmi důležitá vlastnost je rychlost. MySQL je téměř ve všech kategoriích nejrychlejší.

[10]Transakce je uspořádaná skupina databázových operací (dotazů, procedur), která se vnímá a provádí jako jediná jednotka, a to celá, nebo vůbec ne. Nikdy nesmí nastat případ, kdy se vykoná jen její část.

- MySQL je portována na většinu operačních systémů.
- MySQL podporuje přístup z mnoha programovacích jazyků (C, C++, Eiffel, Java, Perl, PHP, Python a Tcl).
- Poměrně dynamický vývoj MySQL. V blízké budoucnosti je plánováno přidání možnosti tvorby kurzorů, aktivačních procedur (triggerů), podvýběrů, pohledů i omezujících podmínek.
- MySQL je standardně zdarma.
- MySQL je dnes velmi rozšířená. Obrovská výhoda vyplývající z rozšířennosti je uživatelská podpora.

Základní typy tabulek

MyISAM – je základní typ tabulky, a je zároveň i výchozí. MyISAM je vylepšená verze nejstarší tabulky ISAM. ISAM je zastaralá a v dnešní době se už nepoužívá. Mezi nejdůležitější vlastnosti MyISAM patří používání komprimovaných indexů, které urychlují SELECTy a odlehčují systémovým prostředkům. Umí všechny základní funkce velmi rychle, z pokročilých funkcí si poradí například s fulltextovým vyhledáváním. Na úkor rychlosti však nepodporuje transakce a cizí klíče.

InnoDB – je druhý základní typ tabulky. Tento typ tabulky se hodí pro projekty s vysokým podílem akčních dotazů oproti dotazům výběrovým. InnoDB umí cizí klíče i transakce. Z toho vyplývají větší nároky na server než u MyISAM, ovšem na druhou stranu cizí klíče usnadňují implementaci a transakce velkou měrou pomáhají zachovat konzistenci naší databáze. Bohužel u InnoDb nelze použít fulltextové vyhledávání (ale dá se to obejít). Výhodou pro administrátory serverů je široké množství nastavení.

Další méně používané typy tabulek jsou MERGE, MEMORY (HEAP), BDB, FEDERATED, ARCHIVE, CSV, BLACKHOLE a nově vyvíjený Falcon. Další informace o typech tabulek je možné najít v dokumentaci [16].

2 NÁVRH SOFTWAROVÉHO ŘEŠENÍ

Poštovní servery jsou systémy v síti (Internetu), zajišťují komunikaci určité skupiny osob prostředníctvím elektronické pošty. Jsou tvořeny hardwarovou a softwarou částí. Tato část práce se bude zabývat návrhem softwarové části poštovního severu. Ta představuje spolupráci několika různých softwarových produktů, kdy každý zastává jinou specializovanou funkci. Jelikož softwaru pro realizaci poštovního serveru je poměrně velké množství, zde popisované řešení je pouze jedním z mnoha možných. K realizaci poštovního serveru bude využit kvalitní open-source software, který je snadno dostupný na Internetu a distribuován zdarma.

2.1 Operační systém

Operační systém bude tvořit základ celého serveru, proto by měl být stabilní a spolehlivý. Debian GNU/Linux je zvolen především na základě osobních zkušeností s distribucí Kubuntu, která vzešla z Debianu, a tudíž má mnoho společných znaků, jako např. instalační program, balíčkovací systém APT, konfigurační soubory atd. Dalším rozhodujícím kritériem je velký podíl instalací Debianu na serverech, což zaručuje důvěryhodnost.

2.1.1 Debian GNU/Linux

Debian je svobodný operační systém (OS) určený k provozu na mnoha různých typech počítačů. Operační systém se skládá ze základního programového vybavení a dalších nástrojů, kterých je k provozu počítače třeba. Vlastním základem OS je jádro. Jelikož Debian používá jádro Linux a většina základních systémových programů byla vytvořena v rámci projektu GNU, nese systém označení GNU/Linux [5].

Debian je konzervativní distribuce, často nasazovaná na servery pro svou stabilitu a snadnou údržbu. Repozitáře Debianu obsahují přes 18733 tzv. deb-balíčků s předkompilovanými programy a dokumentací, připravených pro snadnou instalaci. O správu balíčků se v Debianu stará balíčkovací systém APT (Advanced Packaging Tool).

Další možností získání softwaru je kompilace a následná instalace ze zdrojových kódů. Tento postup je užitečný, pokud je potřeba daný software v aktuálnější verzi než se nachází v repozitářích.

Systém je vyvíjen ve třech verzích. stabilní (stable), testovací (testing) a nestabilní (unstable). Pro základ poštovní serveru je vhodné použít stabilní verzi, která obsahuje pouze důkladně otestovaný a odladěný software. Nevýhodou stabilní verze

je určitá zastaralost softwaru v porovnání s jinými distribucemi. Současnou stabilní verzí distribuce Debian GNU/Linux je verze 5.0.1 s kódovým označením Lenny.

2.2 Agent MTA

Agent MTA slouží k přenosu zpráv z jednoho serveru na druhý. Pro unixové systémy je k dispozici několik různých MTA, jako například Sendmail, Exim, Postfix. V dnešní době je nejpoužívanějším MTA Sendmail. Jeho oblíbenost pramení především z toho, že je (byl) v mnohých distribucích jako výchozí MTA. Nicméně má některé nedostatky (např. monolitickou strukturu, horší zabezpečení). Alternativou k Sendmailu je Postfix, který přestože byl vyvíjen nezávisle, je se Sendmailem kompatibilní, avšak netrpí jeho nedostatky.

2.2.1 Postfix

Postfix je open source MTA agent. Jeho autorem je Wietse Venema[1]. Postfix je malý, bezpečný, flexibilní a rychlý. Přestože je jeho filozofie naprosto odlišná od Sendmailu, je s tímto programem kompatibilní. Hlavní výhodou Postfixu je jeho jednoduchá instalace (bývá obsažen ve většině distribucí ve formě balíčku) a jednoduchá konfigurace (velmi přehledný a dobře komentovaný konfigurační soubor). Další výhodou je jeho modularita. Díky těmto vlastnostem je Postfix vhodný i v nasazeních, kde jsou požadavky na snadnost instalace a obsluhy.

2.2.2 Charakteristika

Spolehlivost – postfix prokazuje svou sílu zejména při práci ve vysokém zatížení. Mnoho softwarových systémů se například chová nepředvídatelně, když jím dojde paměť nebo diskový prostor. Postfix detekuje takové podmínky a místo zhoršení situace nabídne možnost systému se vzpamatovat. Postfix se všemožnými způsoby snaží chovat stabilně a spolehlivě.

Zabezpečení – postfix vychází z modulárního návrhu a zavádí proti útočníkům několik obranných vrstev. V systému Postfixu funguje princip nejmenších oprávnění. Procesy s vyšším oprávněním nikdy nedůvěřují nepriviligovaným procesům. Také platí, že nepotřebné moduly lze deaktivovat, což vede k jednodušší instalaci a vyšší bezpečnosti.

Výkon – Postfix byl navržen pro dosažení vysokého výkonu, ale s předpokladem, že jeho rychlost neomezí jiné systémy. Speciálními technikami omezuje jak

[1] Dr. Wietse Zweitze Venema (* 1951) je nizozemský programátor a fyzik.

počet nových procesů, které je nutné vytvořit, tak i počet přístupů k systému souborů, jež jsou zapotřebí v rámci zpracování zpráv.

Flexibilita – systém Postfix je složen z různých programů a podsystémů, tento přístup zajišťuje vysokou flexibilitu a pružnost. Všechny části lze upravovat pomocí jednoduchých konfiguračních souborů.

Postfix obvykle funguje okamžitě po instalaci a na rozdíl od Sendmailu není Open relay, to znamená, že nepřevezme jakoukoli zprávu bez ohledu na odesílatele i adresáta. Z hlediska správy a konfigurace je jedním z nejednodušších systémů pro zpracování elektronické pošty.

Jak už bylo řečeno, Postfix vychází z modulárního návrhu, což znamená, že pro práci s poštou se nevyužívá jeden rozsáhlý program, jako je tomu například u Sendmailu, ale několik programů (démonů), které spolupracují. Hlavní démon (řídící) je *master*, ten podle potřeby volá vetšinu ostatních procesů.

Posfix podporuje virtuální domény. Umožňuje použít ověřování pomocí knihovny SASL. V kombinaci s tímto je pak velmi vhodné používat zabezpečení protokolu pomocí TLS.

2.3 Filtrování obsahu

Filtr obsahu je nástroj, který kontroluje záhlaví a tělo zprávy a v závislosti na tom, co nejde, provadí nějakou akci. Mezi nejběžnější filtry patří antivirové (např. Clamav, NOD32, BitDefender aj.), a antispamové (např. Spamassasin, Bogofilter, DSPAM aj.) programy. Filtry mohou zprávy měnit, přesměrovat je, odpovídat na ně nebo je označit pro pozdější zpracování dalším nástrojem. Kromě samotných filtrů se na filtraci zpráv podílí ještě další software, který vytváří rozhraní mezi MTA a samotnými filtry. Jeho úkolem je převzít zprávu od MTA a předat ji ke zpracování samotným filtrům. Po profiltrování je zpráva vrácena zpět agentovi MTA.

Velmi často používanou kombinací programů pro filtraci obsahu je trojice Amavisd-new zajišťující rozhraní mezi MTA a filtry, Spamassasin v roli spamového filtru a Clamav v roli virového filtru. Tato práce se zábyvá řešením pomocí výše zmíněných programů vyjma Amavisu, který bude nahrazen programem Sagator. Ten má oproti amavisu několik zajímavých funkcí, jako je např. tvorba statistik a grafů o došlých, prošlých, zlikvidovaných či jiných zprávách, a především snazší instalaci a správu.

2.3.1 Sagator

Sagator je slovenský GNU/GPL systém, který funguje jako prostředník mezi MTA a filtry obsahu. Je navržen velmi flexibilně, což umožňuje nasazení na jakýkoliv MTA. Je také silně modulární, a tak dává administrátorovi široké konfigurační možnosti. Sagator je napsaný v Pythonu. Podporuje celou řadu spamových i virových filtrů.

V terminologi Sagatoru se filtry nazývají skenery. Kromě virových a spamových skenerů, které Sagator neobsahuje a je tedy nutné je nainstalovat zvlášt, obsahuje program i své vlastní skenery. Všechny skenery je možné mezi sebou různě kombinovat, spojovat, což dává velice široké konfigurační možnosti. Například jako první může být použit virový skener a hned po něm spamový. Dále mohou následovat skenery, které buďto zprávu umístí do karantény (obsahuje-li zpráva virus), neudělají nic, pokud se jedná o korektní zprávu, smažou, pokud jde určitě o spam nebo zapíšou do hlavičky *SPAM*, jedná-li se pravděpodobně o spam.

Hlavni konfigurační soubor Sagatoru se jmenuje *sagator.conf* a jedná se vlastně o zdrojový kód Pythonu. Kromě konfiguračních parametrů obsahuje tento soubor také komentáře autora s nejrůznějšími konfiguračními typy.

Rozdělení skenerů

1. Podle zpracovávaných dat

   - skener proudu (streamscanner) – jako vstup slouží datový proud, data jsou předávána v paměti, což je značně rychlejší než skenování souborů. Skenerem file2stream() je možné převést soubor na proud.
   - skener souboru (filescanner) – pracuje se soubory, je pomalejší než skener proudu, ale pokud je nutné předat zprávu některému externímu skeneru (virový, spamový filtr), je jeho použití nevyhutelné. Proud do souboru je možné převést skenerem stream2file().
   - Kombinovaný skener – je kombinací obou předchozích, na vstupu přijímá soubor, ale vystupuje z něj proud a naopak, případně podporuje oba typy dat na vstupu nebo na výstupu.

2. Podle funkce

   - skutečný skener (realscanner) – skenuje zprávu a informuje o výsledku. Kromě vlastní zprávy nastavuje návratovou hodnotu – buďto prázdný řetězec, což lze interpretovat tak, že zpráva je z pohledu skeneru čistá, nebo název identifikovaného viru či slovo *SPAM*.
   - skenovací rozhraní (interscanner) – tvoří rozhraní mezi základním skenovacím strojem a skutečnými skenery

Přehled vybraných skenerů

`modify_header, add_header` – tyto skenery provádějí modifikaci hlaviček zpráv na základě zadaných parametrů

`quarantine` – skener umístí zprávu do karantény

`deliver` – tento skener zajistí, že zpráva bude beze změny doručena

`log` – logovací skener, z jeho výstupu je pak možné pomocí MRTG (Multi Router Tariffic Grapher) zobrazovat statistiky o provozu serveru

`spamassassind` – spamassassin démon skener

V dokumentaci Sagatoru dostupné z [15] je podrobný přehled všech skenerů.

2.3.2 Spamassasin

Spamassassin je spamový filtr napsaný v Perlu, který používá několik různých technik pro indetifikaci nevyžádáné pošty. Obsahuje pravidla pro vyhledávání textových řetězců typických pro spam a řadu předdefinovaných testů. Pokud se některé pravidlo nebo test při zkoumání dané příchozí zprávy uplatní, je zprávě připočten nebo odečten stanovený počet bodů. Zpráva je označena za spam, pokud skóre dosáhne určené hranice (implicitně nastavené na hodnotu 5). Snížení této hodnoty zbůsobí, že filtr bude přísnější a může tedy označit za spam i korektní zprávu. Zvýšení hodnoty má přesně opačný efekt.

Co je spam

Slovo Spam[2] představuje hromadnou nevyžádanou korespondenci, většinou s reklamním podtextem. Spam je šířen zejména z toho důvodu, že je jeho rozesílání velmi levné a snadné. V dnešní době se netýka pouze e-mailu, ale i dalších internetových služeb jako jsou např. Instant Messaging, diskuzní fóra, chat, atd. Pošta, která není spam, je označována jako ham.

Statické techniky pro identifikaci spamu

Síťové testy – Spamassassin může spolupracovat s několika servery, které shromažďují signatury zpráv označených jako spam některými z mnoha tisíců uživatelů po celém světě. Program zašle serveru kontrolní součet zprávy a dostane odpověď, zda jde o známý spam. SpamAssassin může takto spolupracovat se 3

[2]Používá se též zkratka UBE/UCE (Unsolicited Bulk/Commercial Email).

servery: Vipul's Razor, Pyzor a DCC. Každý server používá vlastní klientský program, který je nutné naistalovat na poštovní server před spuštěním SpamAssassinu.

Důvěryhodné sítě – při analýze zprávy prohlíží SpamAssassin hlavičky „Received" od poslední, zapsané poštovním serverem (relay), na němž běží SpamAssassin, směrem zpět a určuje, zda příslušná adresa je důvěryhodná. Důvěryhodná je poslední relay, celá podsíť typu B (o rozsahu 65 tisíc adres), v níž tato relay leží a privátní sítě (neveřejné adresy). Seznam důvěryhodných adres je možné manuálně rozšířit. Nedůvěryhodné adresy jsou hledány na serverech černých listin (black-lists). Je-li adresa nalezena, skóre zprávy se zvýší.

Analýza zpráv – Spamassassin používá testy pro kontrolu jednotlivých částí zprávy. Testuje hlavičky (header), tělo zprávy bez HTML značek (body), tělo s HTML značkami (rawbody), tělo zprávy bez dekódování MIME částí (full), URI v těle zprávy (uri) a adresy v URI (uridnsbl). Je možné vytvářet i vlastní testy.

Černé a bílé listiny – SpamAssassin používá černé a bílé listiny adres. Pokud je odesílatel nalezen na některé listině, je jeho zprávě zvýšeno nebo sníženo skóre. Adresy na listiny je možné přidávat i ručně.

Učení programu

Kromě statických testů se Spamassassin učí ze všech dříve zpracovaných zpráv a svoje chování přizpůsobuje, aby maximalizoval přesnost rozeznávání spamů. Pro učení využívá Spamasasin dvou metod. První jsou automatické bílé listiny, druhou jsou Bayesovské filtry.

Automatické bílé listiny – automatické bílé listiny (AWL, Auto-Whitelists) jsou založeny, na rozdíl od manuálních, na průměrování. Pokud Spamassin například kontroluje dvě zprávy od stejného odesílatele, kdy první získala skóre 20 a druhá 2, jsou tyto hodnoty pomocí AWL zprůměrovány, takže výsledné skóre druhé zprávy bude zvýšeno na 11. V tomto případě se jedná o autoblacklisting, jenž je založený na spamovské historii. Opačný případ by nastal, pokud by tentýž odesílatel poslal zprávu se skórem 0 a následně zprávu se skórem 7, potom by druhé zprávě bylo skóre sníženo na 3,5 a jednalo by se o autowhitelisting, založený na ne-spamovské historii.

– SpamAssasin využívá systému AWL automatického učení následujícím způsobem. Po každé přijaté zprávě je její skóre přičteno k celkovému skóre odesílatele a je zvýšen čítač jeho zpráv. Průměrné skóre je použito k modifikaci

aktuální zprávy. Rozdíl průměrného skóre a skóre aktuální zprávy je vynásoben váhou a přičten ke skóre aktuální zprávy. Hodnota váhy je nastavitelná v rozsahu 0 až 1 [9].

– implicitně je nastavena hodnota 0,5. Pokud je nastavena vyšší hodnota, bude mít větší význam historické skóre, hodnota 1 znamená, že výsledné skóre zprávy se bude rovnat historickému skóre. Hodnota 0 způsobí, že historické skóre bude ignorováno.

Bayesovské filtry – druhou metodou učení jsou bayesovské filtry. U této metody je nutné programu předložit velké množství jak spamových, tak i korektních zpráv, ze kterých se bude učit. Program bude pracovat tím efektivnější, čím více zpráv mu bude předloženo.

– program si podle předložených výukových zpráv vytvoří databázi symbolů (řetězců délky 3 – 15 znaků) nalezených ve zprávách. Ke každému symbolu si zapíše počet jeho výskytů ve spamu a hamu a čas posledního použití při vyhodnocení zprávy. Symboly, které nebyly použity dlouhou dobu, jsou z databáze vymazány, aby se zvýšila efektivita. V druhé databázi je seznam zpráv, z kterých se program učil.

– při kontrole je zpráva rozdělena na symboly, které jsou hledány v databázi. Podle výsledku je zprávě přiřazena pravděpodobnost, že jde o spam. Bayesovským pravidlům označujícím pravděpodobnost menší než 0,5 je přiřazeno záporné skóre, pro pravděpodobnost větší než 0,5 kladné.

2.3.3 Clamav

Clamav je unixový antivirový balík šířený pod licencí GPL, vyvíjený komunitou antivirových odborníků. Je navržen především pro běh na poštovních serverech. Poskytuje množství nástrojů včetně vícevlákonového démona, skener příkazové řádky a nástroj pro automatické aktualizace. Je schopen detekovat přes 400 000 různých virů, červů a trojských koní. Umožňuje také skenování archivů.

2.4 Ukládání a vyzvedávání zpráv

Všechny přijaté zprávy budou po profiltrovaní ukládány do uživatelských schránek v databázi. Odtud si je budou moci jednotliví uživatelé vyzvedávat pomocí protokolů POP3 nebo IMAP. K tomuto účelu dobře poslouží Dbmail, který bude s MTA agentem sdílet databázi uživatelů, jejich zprávy bude ukládat do databáze a následně je pomocí specializovaných nástrojů poskytovat jejich uživatelům.

2.4.1 Dbmail

Dbmail je balík programů, jež umožňují ukládání e-mailových práv do databáze. Podporuje databázové sytémy MySQL, PostgreSQL a SQLite. Za vývojem tohoto produktu stojí nizozemská spolenost NFG a je šířen pod licencí GPL. Dbmail je použitelný s několika MTA, agenty včetně Postfixu. Dbmail přebírá zprávy od MTA pomocí programu *dbmail-smtp* zkrze pipe (rouru) nebo pomocí programu *dbmail-lmtpd* prostřednictvím protokolu LMTP. Tyto pogramy následně zajišťují uložení zprávy do databáze. Z databáze mohou být zprávy vyzvednuty pomocí *dbmail-pop3d*, který používa protokol POP3 a *dbmail-imapd*, kteý využívá protokol IMAP4Rev1. Z toho vyplývá, že DBmail funguje i jako POP3/IMAP sever.

Hlavní výhody Dbmailu

Rozšiřitelnost – dbmail je stejně rozšiřitelný jako databáze, kterou využívá pro ukládání zpráv. Je teoreticky schopen spravovat milióny účtů. Například čtyři různé servery mohou přistupovat pomocí pop3 démona k jednomu databázovému serveru.

Ovladatelnost – dbmail je založen na databázi a muže být tedy spravován změnami v databázi (např. pomocí PHP, Perlu, SQL příkazů), bez přímého přístupu k systému, na kterém běží.

Rychlost – dbmail používá velmi efektivní databázové dotazy k získání informací o přijaté zprávě, což je mohem rychlejší než procházení souborového systému.

Bezpečnost – bezpečnost dbmailu výchází z bepečnosti samotné databáze. Dbmail nijak nezasahuje do souborového systému a nijak neovlivňuje běh dalších spuštěných programů.

Flexibilita – změny v systému (přidávání uživatelů změna hesel atd.) se promítnou okamžitě. Uživatelé mohou být uloženi v databázi nebo vedeni odděleně na LDAP serverech jako je OpenLDAP nebo Active Directory.

3 REALIZACE MAIL-SERVERU

Poslední část této práce se bude zabývat praktickou realizací kompletního poštovního serveru. Především půjde o instalaci a následnou konfiguraci výše popsaného softwaru, dále pak otestováním jednotlivých služeb i celého systému.

Výkon hardwarového vybavení serveru je závislý na zátěži, které bude v provozu vystavován. Pro potřeby této práce postačí starší počítač disponující procesorem Intel Celeron s frekvencí 566 MHz, operační pamětí 320 MB, ethernetovou 10BaseT síťovou kartou a intergovou grafickou kartou Intel.

3.1 Instalace základního systému

Operační systém Debian GNU/Linux je možné stáhnout z jeho domovských stránek [5]. Existuje několik variant, od malých souborů na usb disk nebo disketu pro síťovou instalaci, až po kompletní systém na několika DVD. Pro serverovou instalaci, u které se předpokládá dostupné připojení k Internetu, je například vhodné zvolit malý soubor s ISO obrazem (40 MB) určeným pro zápis na CD, jenž obsahuje pouze základní část Debianu potřebnou ke spuštění instalace, instalační program poté stahuje a instaluje aktuální bálíčky přímo z repozitářů. Výhodou je, že po instalaci je systém plně aktuální a nemusí se provádět dotatečná aktualizace.

3.1.1 Spuštění instalace

Před zahájením samotného instalační procesu je nutné upravit pořadí prohledávání zaváděcích oddílů tak, aby byl zavaděč systému hledán nejprve na CD médiu, to znamená nastavit jako první v pořadí optickou mechaniku. Toto nastavení se provadí v programu SETUP, který je možné spustit stlačením klávesy DEL při startu počítače. Je-li vše nastaveno správně, dojde po vložení CD do mechaniky a následném restartu systému k zahájení instalace. Při zavádění instalačního programu se objeví grafická obrazovka s logem Debianu a nabídkou, z níž je možné vybrat jednu z následujících položek[1]:

`Install` – spustí instalaci v textovém režimu.

`Graphical install` – spustí instalaci v grafickém režimu.

`Advanced options` – otevře další nabídku, kde je možné zvolit pokročilé způsoby instalace, jako je expertní režim, záchranný režim a automatizovaná instalace.

[1] Instalační volby a pokyny jsou podrobně popsány v instalační příručce [6].

`Help` – zobrazí první obrazovku nápovědy se stručným přehledem nápovědných obrazovek.

Instalace v grafickém režimu se odlišuje od textové pouze v podpoře myši, jinak jsou si oba režimy instalace naprosto rovnocenné. Zvolíme tedy výchozí *Install*, čímž se spustí instalace v textovém režimu.

3.1.2 Průběh instalace

Instalační proces se skládá z několika kroků a probíhá téměř automaticky jen s malou účastí uživatele. Délka instalace je silně závislá na výkonu počítače a množství instalovaného softwaru.

Nejprve je uživatel vyzván k výběru jazyka a rozložení klávesnice, poté instalátor detekuje hardware a provede automatickou konfiguraci sítě pomocí DHCP, pokud se operace nezdaří, nabídne ruční konfiguraci. Následně je uživatel vyzván, aby zadal jméno počítače a název domény[2].

Dalším krokem je rozdělení pevného disku a volba souborového systému. Zde je možné nastavit velikost diskových oblastí, včetně výběru souborového systému, který na nich bude použit a přiřadit jim přípojné body z adresářové struktury. Obvykle se vytváří oddělené oddíly pro `/`, `/boot`, `/home` a `swap`. Opět je možné tuto operaci provést ručně nebo ji nechat zcela na instalátoru. Dále lze nastavit šifrování vybraných oddílů a softwarový RAID.

Po vytvoření a naformátování diskových oddílů započne instalace základního systému, kdy se stahují a instalují zakladní systémové balíčky včetně jádra. Tato část instalace je zcela v režii instalátoru a uživatel do ní nijak nezasahuje.

Pokud se instalace základního systému zdařila[3], následuje nejprve vytvoření účtu superuživatele ROOT, kdy je nutné zadat a následně potvrdit heslo pro tento účet a poté vytvoření účtu prvního uživatele. Zde je uživatel vyzván k zadaní jména, loginu a hesla.

Předposledním krokem je instalace připravených softwarových úloh, ty umožní rychlé přizpůsobení počítače pro danou úlohu. Úlohy představují různé činnosti, které je možné s počítačem provádět. Například *desktopové prostředí*, *webový server*, *tiskový server*, *poštovní server*, *standardní systém* atd. Jelikož software pro poštovní server byl již vybrán a žádná další úloha není potřeba, je postačující zvolit pouze možnost *standardní systém*.

Nakonec zbývá nainstalovat zavaděč systému – GRUB. Je možné jej nainstalovat standartně do hlavního zaváděcího záznamu (MBR) prvního disku, nebo zvolit jiné umístění.

[2]Správné nastavaní těchto parametrů je důležité pro konfiguraci Postfixu.

[3]Nezdaří-li se některý bod instalece, je možné jej přeskočit a později se k němu vrátit.

Po dokončení instalace je uživatel vyzván k vyjmutí instalačního média, následuje restart do nově nainstalovaného operačního systému.

3.1.3 Příhlášení do systému

Pro práci v Debianu je nutné se nejprve přihlásit. Po naběhnutí systému se zobrazí přihlašovací dialog a čeká se na zadaní přihlašovacích údajů. Jelikož operace, které budu prováděny při instalaci a konfiguraci systému, přesahují práva přidělená běžnému uživateli, je nutné přihlásit se jako superuživatel ROOT. Jako *login* tedy zadáme *root* a jako *Password* heslo, které jsme zadali při instalaci. Nyní je příkazový interpretr – Bash[4] připraven na zadávání příkazů. Znak `#` za názvem počítače symbolizuje příkazový interpretr superuživatele, shell běžného uživatele bývá označován obvykle znakem `$`.

3.2 Instalace softwaru

Pro instalaci bude využito nástrojů balíčkovacího systému APT, které zajístí jak stažení zvoleného balíčku přímo ze zdrojů (repozitářů), tak i jeho instalaci a ve většíně případů i základní konfiguraci.

3.2.1 Úprava zdojů softwaru

Program Sagator není obsažen v oficiálních zdrojích, proto je před instalací bálíčků nutné nejprve editovat konfigurační soubor programu APT, kde jsou uloženy zdroje softwaru, ze kterých se budou balíčky stahovat a následně instalovat a přidat do něj zdroje Sagatoru. K upravě textových konfiguračních souborů je možné využít některý z široké palety textových editorů, například editor Nano.

```
# nano /etc/apt/sources.list
```

Na konec souboru přídáme následující řádky, čímž přidáme testovací[5] repozitář Sagatoru do seznamu zdrojů. Změny uložíme stiskem kláves *ctrl+o*, editor opustíme stiskem *ctrl+x*.

```
# development (testing) packages
deb http://www.salstar.sk/pub/sagator/debian lenny testing
deb-src http://www.salstar.sk/pub/sagator/debian lenny testing
```

[4]Bash je základní linuxový interpretr příkazů naprogramovaný v rámci projektu GNU.

[5]Je nutné použít testovací repozitář kvůli kompatibilitě s nejnovější verzí antivirové programu Clamav.

Zdroje Sagatoru jsou „podepsané“, proto je nutné stáhnout a uložit GPG klíč, který ověřuje pravost daného zdroje a chrání systém před zneužitím útočníkem.

```
# wget -O - -q http://www.salstar.sk/pub/sagator/SAGATOR-GPG-KEY
 | apt-key add -
```

Aby se provedené změny projevily je nutné aktualizovat seznam dostupných balíčků.

```
# apt-get update
```

Pro jistotu je možné zkontrolovat, jsou-li všechny nainstalované balíčky aktuální a případně je aktualizovat.

```
# apt-get upgrade
```

3.2.2 Instalace jednotlivých softwarových balíčků

V tuto chvíli je možné začít instalovat všechen potřebný software. K tomuto účelu poslouží příkaz `apt-get install <parametr>`, kde <parametr> představuje název balíčku. Příkaz provede jak stažení, tak i nainstalování balíčku.

SSH

Pro správu serveru ze vzdáleného počítače pomocí služby ssh je třeba nainstalovat následující balíčky.

```
# apt-get install openssh-server openssh-client
```

MySQL

Jelikož uživatelé i jejich zprávy budu uloženy v databázi MySQL, je třeba nainstalovat databázový server.

```
# apt-get install mysql-client mysql-server
```

Při instalaci se objeví dialogové okno pro zadání hesla superuživatele pro přístup k databázovému serveru.

Postfix

Jako MTA bude sloužit porgram Postfix. Je nutné jej nainstalovat s podporou databáze MySQL.

```
# apt-get install postfix postfix-mysql
```

Dbmail

Jako úložiště zpráv a POP3/IMAP server bude využit Dbmail. Opět je nutné jej nainstalovat s podporou databáze MySQL.

```
# apt-get install dbmail dbmail-mysql
```

Filtry obsahu

Před instalací Sagatoru je vhodné nejprve instalovat Clamav a Spamassassin, aby je Sagator detekoval a překopíroval potřebné soubory do *CHROOTu*[6], ve kterém „poběží“.

```
# apt-get install clamav
# apt-get install spamassassin spamc
# apt-get install sagator
```

SASL

Aby fungovalo SMTP ověřovaní uživatelů pomocí SASL, je nutné nainstalovat následující balíčky.

```
# apt-get install libsasl2-modules-sql libgsasl7
 libauthen-sasl-cyrus-perl
```

Stunnel

Přenos pošty protokoly POP3 a IMAP bude šifrován pomocí programu Stunnel.

```
# apt-get stunnel
```

Nyní je naistalován všechen potřebný software k realizaci poštovního serveru a je možné přejít ke konfiguraci jednotlivých částí.

3.3 Konfigurace

Nejdůležitejším krokem při realizaci poštovního serveru je správné nastavení jeho dílčích částí.

[6]Change root – změna kořenového adresáře. Pomocí této funkce (a stejnojmenného příkazu) je možné vytvořit uzavřené prostředí, ve kterém poběží potenciálně nebezpečné programy.

3.3.1 Databáze Mysql

Domény a uživatelé, pro které bude server přijímat zprávy budou vedeni v databázi, stejně tak i zprávy jednotlivých uživatelů se budou ukládat do databáze. Je potřeba tedy vytvořit databázi pro Postfix, ve které budou uloženy domény a databázi pro Dbmail, v níž budou vedeni uživatelé a zároveň se do ní budou ukládat i jejich zprávy. Jelikož uživatelé vedeni v databázi Dbmailu budou ti samí, pro které bude postfix přijímat zprávy, není potřeba vytvářet zvlášť tabulku uživatelů pro Postfix a zvlášť pro Dbmail, navíc Dbmail obsahuje nástroj pro snadnou správu uživatelských účtů.

Pro vytvoření databáze Dbmailu poslouží níže uvedený příkaz, za kterým následuje požadavek na zadání hesla superuživatele pro přístup k databázovému serveru.

```
# mysqladmin create <jméno> -u root -p
```

Parametr <jméno> reprezentuje název databáze (např. dbmail).

Dále je nutné přiřadit práva uživateli, který bude s touto databází pracovat. Nejprve je ale nutné přihlásit se k databázovému serveru. Opět bude uživatel vyzván k zadání hesla superuživatele.

```
#  mysql -u root -p
```

Následující příkaz přidělí uživateli dbmail[7] všechna práva nad databází dbmail.

```
> GRANT ALL ON dbmail.* to dbmail@localhost identified by '<heslo>';
```

Parametr <heslo> představuje heslo uživatele dbmail pro přístup k databázovému serveru (např. dbmail). Nyní je možné se odhlásit od databázového serveru.

```
> quit;
```

V tuto chvíli je databáze připravena a je možné vytvořit potřebné tabulky. To lze v Debianu zajistit následujícím příkazem.

```
zcat /usr/share/doc/dbmail-mysql/examples/create_tables.mysql.gz
 | mysql -u dbmail dbmail -p
```

Postfix

Zcela analogicky je možné vytvořit databázi Postfixu (řetězec dbmail nahradíme řetězcem postfix), avšak potřebné tabulky je již nutné vytvořit ručně. Tabulku v níž budou uloženy domény je možné vytvořit níže uvedeným způsobem. Nejprve je nutné se přihlásit k databázovému serveru.

```
#  mysql -u postfix -p
```

[7] Uživatel dbmail byl automaticky vytvořen při instalaci Dbmailu.

Po zadání hesla uživatele postfix je třeba vybrat databázi, do níž bude tabulka přidána.

```
use postfix;
```

Po provedení nasledujícího příkazu se vytvoří tabulku `domains`.

```
CREATE TABLE ‘domains‘ (
‘pkid‘ smallint(6) NOT NULL auto_increment,
‘domain‘ varchar(120) NOT NULL default ’’,
‘transport‘ varchar(120) NOT NULL default ’virtual:’,
‘enabled‘ tinyint(1) NOT NULL default ’1’,
PRIMARY KEY (‘pkid‘)
) ;
```

V tuto chvíli je vytvořen nezbytný databázový základ a je možné přejít ke konfiguraci jednotlivých programů.

3.3.2 Postfix

Konfigurace se provadí pomocí prostých textových souborů a vyhledávacích tabulek. Jednotlivé parametry se zapisují v následujícím tvaru:

```
parametr = hodnota
```

Je možné je zapisovat v libovolném pořadí. Pokud je parametrů více, odělují se čárkami, mezerami, tabulátory nebo novými řádky. Pokračuje-li některý řádek novým řádkem, je třeba začít jej mezerou nebo tabulátorem. Komentář začíná znakem `#`.

Nejdůležitějšími konfiguračními soubory Postfixu jsou *master.cf* a *main.cf*. Soubor *master.cf* obsahuje jeden řádek pro každou službu nebo transport Postfixu. Každý řádek má sloupce specifikující, jak mají jednotlivé programy běžet jako součásti celého systému Postfix [7].

Soubor *main.cf* je jádrem celého Postfixu. Vetšina konfiguračních změn se provadí právě zde. Je možné zde definovat téměř tři sta různých parametů Postfixu.

Pro uložení dalších konfiguračních parametrů používá Postfix standardně vyhledavácí tabulky. Ty mají obvykle formát unixových databázových souborů. V tomto řešení však budou tyto vyhledávací tabulky nahrazeny externí MySQL databází.

Základní konfigurace

Základní konfigurace Postfixu závisí především na doméně, ve které se stroj nachází, a jeho poloze v sítí.

Editací souboru *main.cf* je možné začít konfiguraci.

```
# nano /etc/postfix/main.cf
```

Paremetr `myhostname` musí být nastaven na plně kvalifikovaný název hostitele systému. Pokud byli při instalaci Debianu zadány parametry název počítače a domény, bude parametr automaticky nastaven na hodnotu <`název`>.<`doména`>.

```
myhostname = mail.mojedomena.cz
```

Dále je nutné nastavit parametr `mydomain`. Hodnotou bývá jméno domény, pokud není parametr zadán, je nastaven automaticky z parametru myhostname.

```
mydomain = mojedomena.cz
```

Parametr `myorigin` označuje doménu, která se přidá k adrese odchozích zpráv.

```
myorigin = $mydomain
```

Paremetr `mydestination` určuje, pro které domény bude Postfix na tomto stroji přijímat poštu. To se týká pouze uživatelů se zřízeným účtem v systému. Jelikož server bude přijímat poštu pouze pro domény uložené v databázi (virtuální), ponechá se tento parametr bez hodnoty

```
mydestination =
```

Dalším paremetrem je `relayhost`. Není-li zadána žádná hodnota server posílá zprávy do internetu sám. Pokud je zadána hodnota, většinou SMTP server poskytovatele připojení k internetu (ISP), jsou všechny odchozí zprávy předávány jemu.

```
relayhost = smtp.mujisp.cz
```

Posledním důležitým parametrem je `mynetworks`, určuje IP adresy nebo adresy sítě, kterým bude povoleno odesílat zprávy zkrze server. To zajišťuje, aby server nebyl tzv. *open relay*. Pokud není nutné, aby server postupoval zprávy jiným systémům, je možné parametrem `mynetwork_style`[8] definovat odesílaní pouze z lokálního počítače. Pokud jsou zadány oba parametry, má přednost `mynetworks`.

```
#mynetworks = 192.168.15.32/26
mynetwork_style = host
```

Informace o nastavení dalších parametrů lze získat v [7]. Kompletní konfigurační soubory *main.cf* a *master.cf* jsou uvedeny v příloze D respektive E.

[8]Parametr `mynetwork_style` může kromě hodnoty `host` dále nabývat hodnot `class` a `subnet`

Nastavení přístupu do databáze

Aby Postfix „věděl“, kde a jak má hledat uživatele a domény, pro které bude poštu přijímat, je nutné přidat do souboru *main.cf* příslušné mapy a jim přiřadit mapovací soubory. Ty obsahují informace o nastavení, jež určuje, jak získat požadovanou hodnotu z databáze. Namapování tabulek s doménami a uživateli se zajistí přidáním následujících řádků do souboru *main.cf*.

```
#vyhledavani domen
virtual_mailbox_domains = mysql:/etc/postfix/mysql_domains.cf
#vyhledavani uzivatelu
virtual_mailbox_maps = mysql:/etc/postfix/mysql_mailbox.cf
```

Dále je nutné vytvořit mapovací soubor pro domény

```
# nano /etc/postfix/mysql_domains.cf
```

a vložit do něj následující řádky.

```
user = postfix
password = postfix
dbname = postfix
table = domains
select_field = domain
where_field = domain
hosts = 127.0.0.1
additional_conditions = and enabled = 1
```

Význam jednotlivých položek, jak je uveden v [7]:

`user` – název účtu, který má být použit pro přihlášení do databáze MySQL.

`password` – heslo, které má být použito pro přihlášení do databáze MySQL.

`dbname` – název databáze, která má být použita pro dotaz.

`table` – název tabulky, která má být použita pro dotaz.

`select_field` – název sloupce, který obsahuje hledanou hodnotu.

`where_field` – název sloupce, který obsahuje název klíče.

`additional_conditions` – další porovnání pro klauzoli WHERE příkazu SQL, vytvářeného Postfixem.

Postup pro vytvoření mapovacího souboru pro uživatele je obdobný jako pro domény.

```
# nano /etc/postfix/mysql_mailbox.cf
```

Formát zápisu je ale poněkud odlišný.

```
user = dbmail
password = dbmail
hosts = 127.0.0.1
dbname = dbmail
query =  SELECT alias FROM dbmail_aliases WHERE alias='%s'
 UNION SELECT userid FROM dbmail_users WHERE userid='%s';
```

Nyní už Postfix „ví“, kde má hledat domény a uživatele, pro které bude přijímat poštu. Aby mohly být přijaté zprávy uloženy, je třeba ještě nastavit úložiště zpráv.

3.3.3 Dbmail

Konfigurační soubor Dbmailu *dbmail.conf* je velmi přehledný a dobře okomentovaný. Pro správnou funkci programu je nutné nastavit několik základních parametrů týkajících se především přístupu k databázi. Syntaxe je obdobná jako u Postfixu. Editaci souboru *dbmail.conf* je možné zahájit následujícím příkazem.

```
# nano /etc/dbmail/dbmail.conf
```

Uvedené parametry je třeba nastavit, případně upravit.

Ovladač pro přístup do databáze.

```
driver                = mysql
```

Autentifikační ovladač.

```
authdriver            = sql
```

Počítač, na kterém se nachází databáze.

```
host                  = localhost
```

Soket pro přístup do databáze.

```
sqlsocket             = /var/run/mysqld/mysqld.sock
```

Nazév účtu, který má být použit pro přihlášení do databáze.

```
user                  = dbmail
```

Heslo, které má být použito pro přihlášení do databáze.

```
pass                    = dbmail
```

Název databáze, se kterou se má pracovat.

```
db                      = dbmail
```

Nástavení kódování – musí být shodné s kódováním tabulek v databázi.

```
encoding                = latin1
```

Nástavení výchozího kódování pro zprávy s nerozpoznaným kódováním.

```
default_msg_encoding = latin1
```

Po provedení a uložení změn je nutné Dbmail restartovat.

```
/etc/init.d/dbmail restart
```

Aby byly Postfixem přijaté zprávy předávány Dbmailu, je třeba přidat do souborů *main.cf* a *master.cf* několik parametrů.

```
# nano /etc/postfix/main.cf
```

Parametr `virtual_transport` určuje transport, jenž bude použit pro doručování zpráv na adresy virtuálních schránek. Pro transport bude využit LMTP démon Dbmailu, který naslouchá na portu 24[9].

```
virtual_transport = dbmail-lmtp:[127.0.0.1]:24
```

Dále je nutné editovat soubor *master.cf*

```
# nano /etc/postfix/master.cf
```

a přidat do něj následující řádek, který zajistí, že démon *master* provede transport.

```
dbmail-lmtp     unix    -       -       n       -       -       lmtp
```

Po uložení změn je potřeba restartovat Postfix.

```
# postfix reload
```

a spustit lmtp démona dbmailu.

```
# dbmail-lmtpd
```

[9]Port lze změnit v konfiguračním souboru Dbmailu.

Dbmail pracuje nejen jako úložiště zpráv, ale také jako POP3/IMAP server. Aby bylo možné vyzvedát poštu srkze tyto protokoly, je nutné spustit příslušné démony.

```
# dbmail-pop3d
# dbmail-imapd
```

Pro automatické spuštění démonů po startu počítače je nutné příslušné démony přidat do souboru *dbmail* v adresáři `/etc/default`.

```
# /etc/default/dbmail
```

Syntaxe je následující.

```
START_IMAPD=1
START_POP3D=1
START_LMTPD=1
```

V tomto stádiu konfigurace už je server schopen odesílat a přijímat zprávy pro uživatele a domény uložené v databázi[10]. Zatím ale nijak nekontroluje, jestli zpracovávaná zpráva je spam nebo neobsahuje-li virus. Pro je třeba ještě nakonfigurovat filtry obsahu.

3.3.4 Filtry obsahu

Spamassassin

Po instalaci je kontrola spamu pomocí Spamassassinu deaktivována. Aktivovat ji lze editací souboru *spamassassin*

```
# nano /etc/default/spamassassin
```

a úpravou prametru `ENABLED`.

```
ENABLED=1
```

Konfigurace Spamassassinu je poměrně použitelná hned po instalaci, případně lze vygenerovat konfigurační soubor nový, pomocí webového konfiguračního generátoru na adrese www.yrex.com/spam/spamconfig.php. Vygenerovaný kód je třeba vložit do souboru */etc/mail/spamassassin/local.cf*.

Nejdůležitějším parametrem je `required_score`. Určuje kolika bodů musí zpráva při testech dosáhnout, aby byla označena jako spam. Výchozí hodnota je `5`.

Aby Spamassassin fungoval spolehlivě, je dobré jej naučit, které zprávy má považovat za spam a které za ham. Toho lze docílit následujícími příkazy.

```
# sa-learn --showdots -C /etc/spamassassin --spam /adresar/se/spamem/
# sa-learn --showdots -C /etc/spamassassin --ham /adresar/s/hamem/
```

Pro efektivní učení je třeba, aby každý adresář obsahoval minimálně 200 zpráv.

[10] Přidávání uživatelů a domén bude popsáno v kapitole 3.4

Clamav

Clamav po instalaci nevyžaduje téměř žádné zásahy do konfigurace. Jediný parametr, který je vhodné upravit, je, jak často bude aktualizační démon *freshclam* kontrolovat, je-li virová databáze aktuální. Ve výchozím nastavení tak činí každou hodinu, což je zbytečné. Parametr lze upravit editací souboru *freshclam.conf*

```
# nano /etc/clamav/freshclam.conf
```

a úpravou parametru `Checks`, který udává, kolikrát za den dojde ke kontrole aktualizací.

```
Checks 6
```

Zadáním hodnoty `6`, bude kontrola prováděna každé 4 hodiny.

Sagator

Sagator je mocný nástroj a při konfiguraci dává uživateli velkou volnost. Výchozí konfigurační soubor je poměrně rozsáhlý a většinu jeho obsahu představují komentáře s příklady použití jednotlivých skenerů, kdy pro jejich použití stačí pouze odkomentovat příslušný řádek. Na adrese www.salstar.sk/sagator/examples/ jsou k dispozici příklady konfiguračních souborů, které mohou posloužit jako inspirace[11].

Kompletní konfigurační soubor je přiložen v příloze F a zajišťuje, že „čisté“ zprávy projdou beze změny, zprávy infikované viry budou uloženy do karantény, stejně tak jako zprávy obsahující spustitelné `*.exe` soubory. Zprávy jednoznačně identifikované jako spam budou smazány, zprávám, které budou „na hraně“, bude do hlavičky přidám text ve tvaru `SPAM *****`, kde počet hvězdiček vyjadřuje skóre, které zprávě přidělil Spamassassin. Konfigurační soubor Sagatoru *sagator.conf* se nachází v adresáři `/etc`.

Aby vůbec skenování zpráv proběhlo, musí o Sagatoru vědet Postfix, který mu bude zprávy ke skenování předávat a proskenované zprávy zase příjímat. Do souboru *main.cf* je nutné přidat parametr `content_filter`, jenž definuje pomocí jaké služby a na jakém portu bude zpráva předána k filtraci.

```
content_filter = smtp:[127.0.0.1]:27
```

Dále je potřeba do souboru *master.cf* přidat několik řádků, aby došlo k navrácení již profiltrované zprávy.

[11] Konfigurační soubor vytvořený Davidem Zejdou, jeho seriál o Sagatoru na serveru *root.cz* [19] a dokumentace Sagatoru [15] byly hlavními informačními zdroji při vytváření konfiguračního souboru.

```
127.0.0.1:26    inet n - n - 30 smtpd
 -o content_filter=
 -o myhostname=sagator.mojedomena.cz
 -o local_recipient_maps=  -o relay_recipient_maps=
 -o mynetworks=127.0.0.0/8  -o mynetworks_style=host
 -o smtpd_restriction_classes=  -o smtpd_client_restrictions=
 -o smtpd_helo_restrictions=  -o smtpd_sender_restrictions=
 -o smtpd_recipient_restrictions=permit_mynetworks,reject
```

Aby se provedené změny projevily, je nutné restartovat Postfix.

```
# postfix reload
```

Nyní je server kompletně funkční a zbývá pouze doplnit ověřování a šifrovaní.

3.3.5 Ověřování a Šifrovaní

Cyrus SASL

Ověřování pomocí knihovny Cyrus SASL zajistí, že bude-li chtít někdo odeslat poštu skrze server, musí se nejdříve identifikovat svými přihlašovavími údaji. Konfigurace se provádí pomocí souboru *smtpd.conf*.

```
# nano /etc/postfix/sasl/smtpd.conf
```

Do souboru vložíme nasledujcí řádky.

```
pwcheck_method: auxprop
auxprop_plugin: sql
sql_engine: mysql
mech_list: DIGEST-MD5 CRAM-MD5 PLAIN LOGIN
sql_hostnames: 127.0.0.1
sql_user: dbmail
sql_passwd: dbmail
sql_database: dbmail
sql_select: SELECT passwd FROM dbmail_users WHERE userid = '%u'
```

Význam jednotlivých položek:

`pwcheck_method` – určuje metodu ověřovaní, parametr `auxprop` říká, že bude použit externí soubor s hesly SASL.

`auxprop_plugin` – název exteního pluginu, který se použije k získání hesla.

`sql_engine` – typ SQL databáze.

`mech_list` – seznam ověřovacích mechanismů, které budou použity.

`sql_hostnames` – adresa, na které se nachází databázový server.

`sql_user` – název účtu, který má být použit pro přihlášení do databáze.

`sql_passwd` – heslo, které má být použito pro přihlášení do databáze.

`sql_database` – název databáze, která má být použita pro dotaz.

`sql_select` – SQL dotaz.

Dále je nutné přidat do souboru *main.cf* několik omezení a především povolit ověřování [1].

```
smtpd_recipient_restrictions = reject_unauth_pipelining,
        permit_mynetworks, permit_sasl_authenticated,
        reject_non_fqdn_recipient,
        reject_unauth_destination, permit

smtpd_sender_restrictions = permit_sasl_authenticated,
        permit_mynetworks, reject_non_fqdn_sender,
        reject_unknown_sender_domain,
        reject_unauth_pipelining, permit

smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_sasl_path = smtpd
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain =
```

Všechny použité prametry jsou podrobně popsány v [7].

TLS

Konfigurace TLS spočívá především ve vygenerování potřebných certifikátů a následnému povolení TLS v Postfixu. Pro generování certifikátů je možné použít následující sérii příkazů [2].

Nejprve je potřeba se přepnout do adresáře, v němž budou certifikáty uloženy.

```
# cd /etc/postfix/
```

Poté je možné začít generovat potřebné klíče a certifikáty.

```
# openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
```

Bude vyžadováno zadaní hesla pro *smtpd.key*.

```
# chmod 600 smtpd.key
# openssl req -new -key smtpd.key -out smtpd.csr
```

Nyní bude vyžadováno kromě hesla pro *smtpd.key* několik dalších údajů:

- Zkratka země –, `CZ`
- Název státu – `Czech Republic`
- Město – `Brno`
- Název organizace – `Utko`
- Název organizační podjednoky – `mail.mojedomena.cz`
- E-mailová adresa – `moje@adresa.cz`
- Ostatní položky jsou volitelné

Po dokončení operace je možné zadat další příkaz.

```
# openssl x509 -req -days 3650 -in smtpd.csr
 -signkey smtpd.key -out smtpd.crt
```

Opět bude vyžadováno zadání hesla pro *smtpd.key*.

```
# openssl rsa -in smtpd.key -out smtpd.key.unencrypted
```

Znovu heslo pro *smtpd.key*.

```
# mv -f smtpd.key.unencrypted smtpd.key
# openssl req -new -x509 -extensions v3_ca
 -keyout cakey.pem -out cacert.pem -days 3650
```

Bude třeba zadat stejné parametry jako při generování souboru *smtpd.csr*. Po úspěšném provedení operace je vygenerováno vše potřebné a je možné nastavit Postfix aby komunikaci šifroval.

```
smtpd_tls_cert_file = /etc/postfix/smtpd.crt
smtpd_tls_key_file = /etc/postfix/smtpd.key
smtpd_use_tls=yes
smtp_tls_security_level = may
smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
```

Nakonec je nutné v souboru *master.cf* nastavit služby, u nichž bude vyžadována autentizace a šifrování.

```
# nano /etc/postfix/sasl/smtpd.conf
```

Je třeba upravit nebo přidat následující řádky, které zajistí ověřování a šifrování pro služby SMTP Submission (port 587) a SMTPS (port 465).

```
submission inet n       -       -       -       -       smtpd
#  -o smtpd_tls_security_level=encrypt
  -o smtpd_tls_auth_only=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,
     reject_unauth_destination,reject
  -o smtpd_sasl_security_options=noanonymous,noplaintext
  -o smtpd_sasl_tls_security_options=noanonymous
#  -o milter_macro_daemon_name=ORIGINATING
smtps     inet  n       -       -       -       -       smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o smtpd_sasl_security_options=noanonymous,noplaintext
  -o smtpd_sasl_tls_security_options=noanonymous
```

Stunnel

Pomocí Stunnelu bude šifrována komunikace probíhající skrze protokoly POP3S a IMAPS. Podobně jako u TLS i zde je nutné vygenerovat potřebné klíč a certifikát, tenkokrát ale bez hesla. Nejprve je potřeba se přepnout do adresáře, ve kterém bude vše uloženo.

```
# cd /etc/stunnel/
```

Poté je možné začít generovat potřebné soubory.

```
# openssl req -new -x509 -nodes -out cert.pem
 -keyout key.pem -days 1098
```

Opět bude položeno několik dotazů jako při generování certifikátu. Po vygenerování klíče i certifikátu je potřeba změnit jejich přístupová pravá.

```
# chmod 600 cert.pem
# chmod 600 key.pem
```

Nyní je možné přejít k vlastní konfiguraci. Ta se provadí editací souboru *stunnel.conf*.

```
# nano /etc/stunnel/stunnel.conf
```

Nejprve je třeba zadat cestu ke klíči a certifikátu.

```
key = /etc/stunnel/key.pem
cert = /etc/stunnel/cert.pem
```

Dále už je možné nastavit, které protokoly mají být šifrovány, v tomto případě tedy pouze POP3 a IMAP.

```
[pop3s]
accept  = 995
connect = 110

[imaps]
accept  = 993
connect = 143
```

K automatickému spuštění Stunnelu po startu počítače je potřeba editovat jeho spuštěcí skript

```
#nano /etc/init.d/stunnel4
```

a následně vyhledat a upravit parametr `ENABLED`.

```
ENABLED=1
```

Aby se Stunnel vůbec spustil, je ještě nutné upravit soubor *stunnel4* v adresáři `/etc/default`

```
# nano /etc/default/stunnel4
```

a upravit stejný parametr jako ve spouštěcím skriptu.

```
ENABLED=1
```

3.4 Správa uživatelů a domén

V této kapitole bude popsáno, jakým způsobem lze spravovat uživatelské účty a domény, pro které má server přijímat zprávy.

3.4.1 Správa uživatelů

Dbmail poskytuje nástroj *dbmail-users*, jenž usnadňuje správu uživatelských účtů. Přidat uživatele je možné pomocí příkazu

```
# dbmail-users -a <uzivatel> -w <heslo>
```

kde `<uzivatel>` je uživatelské jméno a `<heslo>` heslo pro přihlášení skrze protokol POP3 nebo IMAP při ověřování odesílatele pomocí SASL.

Vytvořenému uživateli je možné přidat libovolný počet e-mailových adres (aliasů) užitím příkazu

```
# dbmail-users -c <uzivatel> -s <jmeno@mojedomena.cz>
```

kde `<uzivatel>` je uživatelské jméno a `<jmeno@mojedomena.cz>` je e-mailová adresa (alias) uživatele. Pokud je potřeba zadat více aliasů, oddělují se znakem čárka (`,`). Je-li potřeba uživateli odstanit některý z jeho aliasů, je možné to provést příkazem

```
# dbmail-users -c <uzivatel> -S <jmeno@mojedomena.cz>
```

význam položek je stejný jako v předchozím případě.

Dále je možné uživateli přesně definovat velikost jeho schránky příkazem

```
# dbmail-users -c <uzivatel> -m <velikost>
```

kde `<velikost>` představuje číselný údaj následovaný jednotkou (*B – byty*, *KB – kilobyty* nebo *MB – megabyty*). Výchozí jednotkou jsou *byty*.

Pokud uživatel vyžaduje přesměrování přichozích zpráv na jinou adresu, je možné to zajistit zadáním příkazu

```
# dbmail-users -x <jmeno@mojedomena.cz> -t <jmeno@jinadomena.cz>
```

kde `<jmeno@mojedomena.cz>` představuje adresu, která má být přesměrována a `<jmeno@jinadomena.cz>` představuje adresu, na kterou bude pošta přeposílána. Zrušení přeposílání lze provést příkazem

```
# dbmail-users -x <jmeno@mojedomena.cz> -T <jmeno@jinadomena.cz>
```

Význam parametrů je stejný při vytvoření přesměrování.

Smazaní uživatele včetně všech jeho položek zajistí příkaz

```
# dbmail-users -d <uzivatel>
```

Více o možnostech nástroje *dbmail-users* je možné najít v jeho manuálových stránkách, které je možné prohlížet po zadání následujícího příkazu.

```
# man dbmail-users
```

Znázornění tabulek Dbmailu

V tabulce *dbmail_users* jsou uloženi uživatelé, jejich hesla a další údaje jako např.: velikost a obsazenost jejich schránky, čas posledního přihlášení apod. Každý uživatel má také přidělen jednoznačný identifikátor (sloupec *user_idnr*).

Tabulka *dbmail_aliases* obsahuje seznam e-mailových adres (sloupec *alias*) a hodnoty (sloupec *deliver_to*), které obsahují identifikátor uživatele, jemuž příslušná adresa patří, případně transport na jinou adresu.

Tabulka *dbmail_mailboxes* obsahuje seznam s názvy adresářů (sloupec *name*) vytvořených serveru a položky (sloupec *owner_id*), které opět jednoznačně identifikují vlastníka adresáře. Dále je zde několik příznakových paremtrů určených pro IMAP komunikaci.

user_idnr	userid	passwd		last_login
1	uzivatel1	heslo1		2009-05-25 19:40:47
2	uzivatel2	heslo2	.....	2009-05-25 18:44:13

Tab. A.1: Zkrácený výpis tabulky *dbmail_users*.

alias_idnr	alias	deliver_to	client_idnr
1	uzivatel1@mojedomena.cz	1	0
2	uzivatel1.firma@mojedomena.cz	uzivatel1.firma@jinadomena.cz	0
3	uzivatel1.privat@mojedomena.cz	1	0
4	uzivatel2@mojedomena.cz	2	0
5	uzivatel2.firma@mojedomena.cz	uzivatel1@mojedomena.cz	0
6	uzivatel2.privat@mojedomena.cz	2	0

Tab. A.2: Výpis tabulky *dbmail_aliases*.

mailbox_idnr	owner_idnr	name	seen_flag	answered_flag		permission
1	1	INBOX	1	1		2
2	2	INBOX	1	1	.....	2
3	2	SPAM	1	1		2

Tab. A.3: Zkrácený výpis tabulky *dbmail_mailboxes*.

3.4.2 Správa domén

Správa domén je poněkud komplikovanější, neboť pro tuto činnost není k dospozici žádný nástroj. Je proto nutné přihlásit se k databázovému serveru a potřebné domény přidat do příslušné tabulky pomocí SQL příkazů. Jelikož tabulka pro domény se náchazí v databázi postfix, je nutné se příhlásit následujícím způsobem.

```
# mysql -u postfix -p
```

Po zadaní hesla je potřeba vybrat databázi, s níž se bude pracovat

```
> use postfix;
```

a následujícím způsobem je možné vložit potřebné domény.

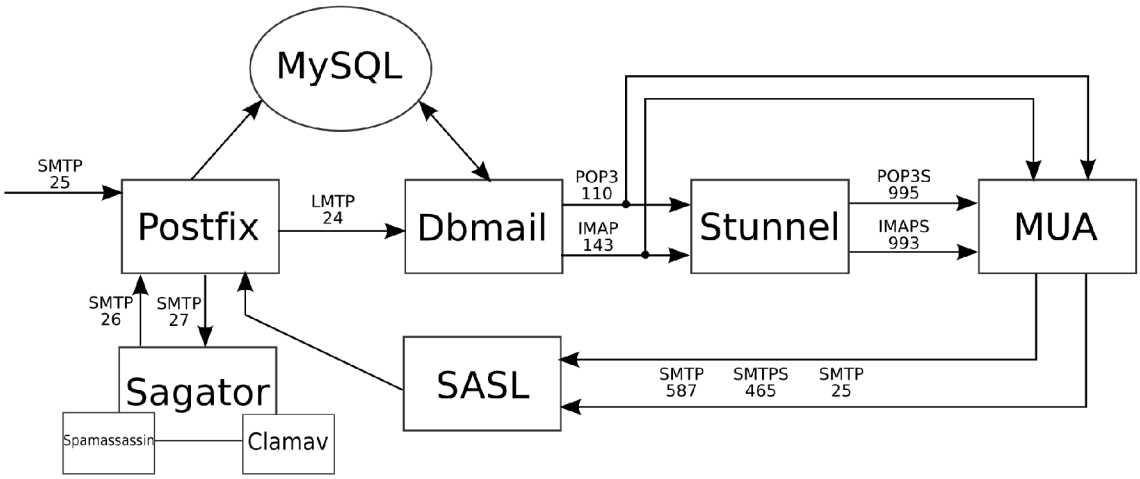
```
> INSERT INTO domains (domain) VALUES
   ('mojedomema.cz'),
   ('dalsidomema.com'),
   ('adalsidomema.org');
```

Pokud je potřeba některou doménu odebrat, je možné použít následující příkaz.

```
> DELETE FROM domains WHERE domain='dalsidomema.com';
```

3.5 Testování funkčnosti

Pro otestování funkčnosti jednotlivých služeb je potřeba mít v databázi přidané uživatele a domény, na které budou testovací zprávy posílány.



Obr. 3.1: Zjednodušené schéma funkce serveru.

K testování je možné použít poštovního klienta (MUA) a v něm postupně nastavovat a testovat jednotlivé služby. Ne vždy je však klientský program po ruce a tak je nutné spokojit se s konzolovým nástroji, jako jsou *telnet* a *openssl*, které budou využity i zde.

Pro lepší přehled o tom, co se při testování děje, a pro snažší detekci chyb je vhodné sledovat logovací soubory. Ty se nachází v adresáři `/var/log`.

Služby mail – serveru

služba	port	šifrování
SMTP	25	ne
POP3	110	ne
IMAP	143	ne
SMTPS	465	ano – SSL
SMTP Submission	587	ano – STARTTSL
POP3S	995	ano – SSL
IMAPS	993	ano – SSL

Tab. 3.1: Přehled služeb podporovaných mail – serverem.

3.5.1 Nešifrované služby

Nešifrované služby je možné testovat například pomocí *telnetu*. Průběhy testování nešifrovaných služeb jsou přiloženy v přílohách. Testování SMTP v příloze A, testování POP3 v příloze B a testování IMAP v příloze C.

3.5.2 Šifrované služby

Pro testování šifrovaných služeb, je možné využít nástroj *openssl* s parametrem *s_client*. Pro protokoly POP3S, IMAPS, SMTPS je použití následující.

```
# openssl s_client -connect localhost:<port> -ssl3
```

Připojení skrze SMTP Submission se provadí přes SMTP port 25, příkaz ale dále obsahuje upřesňující parametry.

```
# openssl s_client -connect localhost:25 -starttls smtp
```

Po úspěšném přihlášení dojde k výpisu bezpečnostního certifikátu a následně je v závisloti na zvolené službě možné zadávat příkazy stejným způsobem jako u přihlášení pomocí *telnetu*.

3.5.3 Antivirus

Jak je patrné z konfiguračního souboru Sagatoru (příloha F), zprávy jsou testovány pomocí knihovny *libclam*, což podle [19] mnohem rychlejší a spolehlivější než testování pomocí démona *clamd*.

Pro otestování správné funkce antivorového filtru je možné použít testovací soubor *Eicar*[12], dostupný na adrese www.eicar.org/anti_virus_test_file.htm, který se vloží jako příloha k odesílané zprávě. K dispozici jsou komprimované i nekomprimované verze souboru. Při testování odhalil Clamav všechny čtyři typy souboru.

3.5.4 Antispam

Testování antispamu na serveru, který není v „ostrém“ provozu, je možné např. pomocí *telnetu*, kdy se pomocí postupu popsaném v příloze A odesílají na server zprávy obsahující řetězce, jež se často vyskytují ve spamových zprávách a podle výsledků se upravují příslušné konfigurační soubory.

[12]Eicar vznikl společným úsilím dodavatelů antivirových programů z celého světa. Jeho smyslem je umožnit uživatelům zjistit reakci antivirových programů a ověřit jejich instalaci bez nebezpečí rozšíření infekce reálného viru.

4 ZÁVĚR

Cílem této práce bylo podrobněji se seznámit problematikou elektronické pošty, získat informace o její architektuře a fungování v prostředí Internetu. Dále pak získat přehled o softwarových produktech, ze kterých jsou v dnešní době složeny poštovní servery a navrhnout a realizovat vlastní řešení poštovního serveru, běžícího na operačním systému GNU/Linux a využívajícího databázi MySQL.

Při výběru softwaru byl kladen důraz především na jeho kvalitu a použitelnost. Jelikož softwaru pro implementaci mail-serveru je v dnešní době velké množství, byly vybrány produkty, jejichž schopnosti a možnosti jsou prověřeny mnoha tísíci uživateli po celém světě. Důležitou roli hrála také dostupnost a kvalita informačních zdrojů, ze nichž bylo možné informace o konkrétním produktu čerpat.

Základ realizovaného poštovního serveru je tvořen stabilní verzí operačního systému Debian GNU/Linux, která nese kódové označení Lenny. O přenos pošty se stará MTA agent Postfix, který má své uživatele uložené v databázi MySQL. Filtraci obsahu provádí trojice produktů. Sagator v roli prostředníka mezi MTA a filtry, Spamassasin v roli spamového filtru a Clamav coby virový filtr. Ukládání zpráv do databáze zajišťuje program Dbmail, jenž s Postfixem sdílí databázi uživatelů. Dbmail rovněž slouží jako POP3/IMAP server. Pro větší bezpečnost uživatelských dat umožňuje server zašifrovat komunikaci mezi poštovním klienten uživatele a serverem samotným. Proti neoprávněnému použití je sever chráněn ověřováním uživatelů.

Obecně lze říci, že vyhledávání dat v databázi je mnohem rychlejší než vyhledávání v souborech uložených na pevném disku. Také správa databáze je mnohem snažší a přehlednější, navíc databázový server může být veden odděleně a sdílet data pro více služeb. Reference Dbmailu hovoří o více než 1100 uživatelských účtech a velikosti databáze přes 200 GB.

V porovnání s komerčními řešeními pro elektronickou poštu (např. Windows Exchange Server) je toto výhodné zejména z finanční stránky. Veškerý použitý software je zdarma a snadno dustupný na Internetu. Dále toto řešení přináší zkušenějšímu uživateli maximální flexibilitu a tím i možnost jej přizpůsobit přesně na míru podmínkám, ve kterých bude provozováno.

Toto řešení je otevřené a je tedy možné do něj integrovat další služby, jako například webové rozhraní pro zprávu databáze, webové rozhraní pro přihlašování uživatelů apod.

LITERATURA

[1] ABRAHAMSEN, I. *How to set up a mail server on a GNU / Linux system* [online]. 2009, Last Update: 2009-05-15 [cit. 20. 5. 2009]. Dostupné z URL: <http://flurdy.com/docs/postfix/edition7.html>.

[2] BURDA, Z. *Mailserver – Postfix, IMAP, Maildrop a MySQL* [online]. 2006, Vloženo 2006/01/13 21:48:45 [cit. 23. 5. 2009]. Dostupné z URL: <http://www.zdenda.com/Mailserver-Postfix-IMAP-Maildrop-MySQL>.

[3] CRISPIN, M. *RFC 3501: INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1* [online]. IETF, Network Working Group, March 2003. [cit. 15. 5. 2009] Dostupné z URL: <http://tools.ietf.org/html/rfc3501>.

[4] Dbmail. *DBMail Wiki* [online]. 2009, Last modified: 2009/05/22 22:18 [cit. 23. 12. 2009]. Dostupné z URL: <http://www.dbmail.org/dokuwiki/doku.php>.

[5] Debian. *Debian – Univerzální operační systém* [online]. 1997-2009, Poslední změna: st, 29. dub 11:30:36 UTC 2009 [cit. 22. 5. 2009]. Dostupné z URL: <http://www.debian.org/index.cs.html>.

[6] Debian. *Debian GNU/Linux – instalační příručka* [online]. 1997-2009, Poslední změna: st, 29. dub 11:30:36 UTC 2009 [cit. 22. 5. 2009]. Dostupné z URL: <http://www.debian.org/releases/stable/i386/index.html.cs>.

[7] DENT, K. *Postfix - kompletní průvodce*. Odpovědný redaktor Martin Kysela; přeložil Ludvík Roubíček. 1. vyd. Praha : Grada Publishing, a.s., 2005. 252 s. ISBN 80-247-1029-3.

[8] KLENSIN, J. *RFC 5321: Simple Mail Transfer Protocol* [online]. IETF, Network Working Group, October 2008. [cit. 18. 5. 2009] Dostupné z URL: <http://tools.ietf.org/html/rfc5321>.

[9] MOUČKA, B. *Vyladťe si svůj SpamAssassin* [online]. Zpravodaj ÚVT MU. ISSN 1212-0901, 2005, roč. XV, č. 5, s. 8-12 [cit. 15. 5. 2009]. Dostupné z URL: <http://www.ics.muni.cz/zpravodaj/articles/334.html>.

[10] MYERS, J., ROSE, M. *RFC 1939: Post Office Protocol - Version 3* [online]. IETF, Network Working Group, May 1996 [cit. 15. 5. 2009]. Dostupné z URL: <http://tools.ietf.org/html/rfc1939>.

[11] PETERKA, J. *SMTP* [online]. 1994 [cit. 16. 5. 2009]. Dostupné z URL: <http://www.earchiv.cz/a94/a444c110.php3>.

[12] PETERKA, J. *Elektronická pošta á la TCP/IP - část I.* [online]. 1998. [cit. 16. 5. 2009]. Dostupné z URL: <http://www.earchiv.cz/a98/a804c200.php3>.

[13] Root.cz. *Historie operačního systému GNU/Linux* [online]. Dostupné z URL: <http://www.root.cz/texty/historie-operacniho-systemu-gnulinux/>.

[14] Root.cz. *Přehled linuxových distribucí* [online]. Dostupné z URL: <http://www.root.cz/texty/prehled-linuxovych-distribuci/>.

[15] ONDREJ, J. *Documentation (for latest development release)* [online]. 2009 [cit. 22. 5. 2009]. Dostupné z URL: <http://www.salstar.sk/sagator/doc.php>.

[16] Sun Microsystems, Inc. *Chapter 13. Storage Engines* [online]. 2008, Posted by Adrian Singer on January 26 2008 1:15pm [cit. 17.5. 2009]. Dostupné z URL: <http://dev.mysql.com/doc/refman/5.0/en/storage-engines.html>.

[17] ŠŤASTNY, P. *Typy tabulek v MySQL* [online]. 27.03.2007 [cit. 17.5. 2009] Dostupné z URL: <http://www.pweb.cz/a/14/typy-tabulek-v-mysql.html>.

[18] Zajíc, P. *MySQL (4) - něco terminologie* [online]. 11.3.2005 15:00 [cit. 17.5. 2009]. Dostupné z URL: <http://dev.mysql.com/doc/refman/5.0/en/storage-engines.html>.

[19] ZEJDA, D. *Seriál Sagator* [online]. 2005 [cit. 20. 5. 2009]. Dostupné z URL: <http://www.root.cz/serialy/sagator/>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

e-mail Electronic Mail (Elektronická pošta)

MTA Mail Transfer Agent (poštovní přenosový agent)

MDA Mail Delivery Agent (agent doručení zprávy)

MUA Mail User Agent (poštovní uživatelský agent)

MS Message Store (úložiště zpráv)

POP3 Post Office Protokol version 3 (poštovní protokol verze 3)

IMAP Internet Message Access Protocol (protokol aplikace internetové pošty)

DNS Domain Name System (systém doménových jmen)

IP Internet Protocol (internetový protokol)

RFC Request For Comments (žádost o komentář)

TLS Transport Layer Security (bezpečnostní přenosová vrstva)

SSL Secure Sockets Layer (vrstva bezpečných socketů)

MIME Multipurpose Internet Mail Extensions (víceúčelové rozšíření internetové pošty)

GPL General Public License (všeobecná veřejná licence)

MIPS Microprocessor without Interlocked Pipeline Stages (procesor bez automaticky organizované pipeline)

APT Advanced Packaging Tool (pokročilý balíčkovací nástroj)

UBE/UCE Unsolicited Bulk/Commercial Email (nevyžádané množství/reklamní pošta)

MBR Master Boot Record (hlavní zaváděcí záznam)

SASL Simple Authentication and Security Layer (vrstva jednoduche autentikace a bezbecnosti)

CD Compact Disc (Kompaktní Disk)

DVD Digital Versatile Disc (Digitální víceúčelový disk)

RAID Redundant Array of Independent Disks (Vícenásobné diskové pole nezávislých disků)

SEZNAM PŘÍLOH

A TESTOVÁNÍ SLUŽBY SMTP

Jelikož server vyžaduje autentizaci uživatele musí se nejprve pomocí následujícího příkazu vygenerovat přihlašovací údaje

```
$ perl -MMIME::Base64 -e 'print
 encode_base64("<uzivatel>\0<uzivatel>\0<heslo>");'
```

kde `<uzivatel>` je uživatelské jméno a `<heslo>` heslo uživatele, skrze něhož bude přihlášení probíhat. Příkaz vygeneruje řetězec znaků kódovaný v base64[1], který se používá při ověřovaní pomocí mechanismu PLAIN. Následně je možné příhlásit se k serveru na SMTP port 25, na kterém není striktně vyžadováno šifrování.

```
$ telnet localhost 25
```

Mělo by dojít k výpisu následujících řádků.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mojedomena.cz ESMTP Postfix (Debian/GNU)
```

Pokračovat je možné představením se.

```
EHLO localhost
```

Odpovědí serveru bude výpis podoporovaných funkcí.

```
250-mojedomena.cz
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN DIGEST-MD5 LOGIN CRAM-MD5
250-AUTH=PLAIN DIGEST-MD5 LOGIN CRAM-MD5
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Nyní je možné se přihlásit použitím vygenerovaného řetězce.

```
AUTH PLAIN <vygenerovany_retezec>
```

[1] Base64 je datový formát zobrazující binární data pomocí tisknutelných znaků ASCII.

Pokud jsou zakódované přihlašovací údaje správné, server potvrdí ověření uživatele.

```
235 2.7.0 Authentication successful
```

Následně je možné začít psát zprávu. Nejprve se uvede adresa odesílatele. Může být libovolná, musí však být ve správném formátu.

```
MAIL FROM: adresa@domena.cz
```

Pokud je formát zprávy zadán správně, sever ji přijme a potvrdí.

```
250 2.1.0 Ok
```

Pokračuje se zadáním příjemce. Aby došlo k přijetí zprávy, musí být zadána adresa některého z uživatelů uložených v databázi.

```
RTCP TO: jmeno@mojedomena.cz
```

Opět dojde k potvrzení.

```
250 2.1.5 Ok
```

Po zadání příkazu `DATA` je možné začít psát vlastní zprávu.

```
DATA
354 End data with <CR><LF>.<CR><LF>
```

Lze uvést předmět (*Subject:*) a některé další údaje viz.[7]. Poté následuje text zprávy.

```
Subject: testovaci zprava

Ahoj toto je testovaci zprava.
.
```

Znak tečka (.) na samostatném řádku znamená ukončení psaní a odeslaní zprávy. Server odpoví, že zprávu přijal a zařadil ji do fronty ke zpracování.

```
250 2.0.0 Ok: queued as 0C6D9A72EF
```

Příkazem `QUIT` je možné ukončit spojení.

```
QUIT
```

Server se rozloučí a ukončí spojení.

```
221 2.0.0 Bye
Connection closed by foreign host.
```

B TESTOVÁNÍ SLUŽBY POP3

Protokol POP3 využívá standardně port 110, nejinak je tomu i v tomto případě. Pro přihlášení k POP3 serveru je možné využít opět telnet.

```
$ telnet localhost 110
```

Po navázání spojení se vypíšou následující řádky.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK DBMAIL pop3 server ready to rock
```

Nejprve je nutné se přihlásit.

```
USER <UZIVATEL>
+OK Password required for pepa
PASS <HESLO>
+OK <UZIVATEL> has 2 messages (3626 octets)
```

Jak je vidět, uživatel má ve schránce dvě zprávy a jejich velikost je 3626 oktetů. Příkazem list je možné si zobrazit podrobnosti o jednotlivých zprávách.

```
LIST
+OK 2 messages (3626 octets)
1 2177
2 1449
.
```

Zobrazení vybrané zprávy (např. zprávy číslo 1) je možné pomocí příkazu `RETR 1`.

```
RETR 1
+OK 1149 octets
Received: from mojedomena.cz (localhost [127.0.0.1])
        by sagator.mojedomena.cz (Postfix)
with ESMTP id A0B3FA72F3
        for <jmeno@mojedomena.cz>;
Thu, 28 May 2009 14:26:32 +0200 (CEST)
X-Spam-Checker-Version: SpamAssassin 3.2.5 (2008-06-10) on
        mojedomena.cz
X-Spam-Level:
X-Spam-Status: No, score=-1.4 required=5.0 tests=ALL_TRUSTED
```

```
autolearn=ham version=3.2.5
X-Sagator-Scanner: 1.2.0-0.beta26 at b05-524d;
        log(status(quarantine(drop(buffer2mbox(libclam()),
        ParseMail(file_type()))))),
        status(rename(cache(SpamAssassinD())*const()),
        drop(rename(cache()>=const())),
        deliver(modify_subject(rewrite_name(cache()))))
X-Sagator-ID: 20090528-142632-0001-02057-cWtldU@b05-524d
Received: from localhost (localhost [127.0.0.1])
        by mojedomena.cz (Postfix)
with ESMTPA id 8E311A72F0
        for <jmeno@mojedomena.cz>;
Thu, 28 May 2009 14:24:07 +0200 (CEST)
Subject: testovaci zprava
Message-Id: <20090528122433.8E311A72F0@b05-524d.kn.vutbr.cz>
Date: Thu, 28 May 2009 14:24:07 +0200 (CEST)
From: adresa@domena.cz
To: undisclosed-recipients: ;
Return-Path: adresa@domena.cz
MIME-Version: 1.0

Ahoj, toto je testovaci zprava.
.
```

Jak je patrné zpráva neobsahuje pouze odeslaná data, ale při zpracování byla doplněna o data nová, které však bývají při čtení pomocí většiny MUA skryta[1]. Jsou zde především informace o průběhu zpracovaní filtry obsahu.

Vybranou zprávu lze smazat příkazem `DELE`.

```
DELE 1
+OK message 2 deleted
```

Ukončení spojení se provadí obdobně jako u SMTP.

```
QUIT
+OK see ya later
Connection closed by foreign host.
```

[1]Program Kmail dokáže „vytáhnout“ ze zprávy informace o zpracování Spamassassinem a graficky je zobrazit.

C TESTOVÁNÍ SLUŽBY IMAP

Protokol IMAP je výrazně složitější než POP3 a umožňuje o mnoho víc funkcí. Standardně využívá port 143. Přihlásit se k IMAP serveru je možné následovně.

```
$ telnet localhost 143
```

Po navázaní spojení se vypíšou následující řádky.

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK dbmail imap (protocol version 4r1) server 2.2.10 ready to run
```

Každý příkaz musí být uveden unikátním idenitfikátorem v rámci daného sezení (tečka před každým příkazem). Přihlásit se je možné pomocí následujícího příkazu.

```
. LOGIN <UZIVATEL> <HESLO>
. OK LOGIN completed
```

Služby, které server podporuje je možné vypsat zadáním příkazu `CAPABILITY`.

```
. CAPABILITY
* CAPABILITY IMAP4 IMAP4rev1 AUTH=LOGIN ACL NAMESPACE
 CHILDREN SORT QUOTA THREAD=ORDEREDSUBJECT UNSELECT IDLE
. OK CAPABILITY completed
```

Pro zobrazení adresářů je třeba použít příkaz `LIST` v následující podobě.

```
. LIST "" "*"
* LIST (\hasnochildren) "/" "INBOX"
. OK LIST completed
```

Přepnutí do adresáře `INBOX`.

```
. SELECT "INBOX"
* 3 EXISTS
* 3 RECENT
* FLAGS (\Seen \Answered \Deleted \Flagged \Draft)
* OK [PERMANENTFLAGS (\Seen \Answered \Deleted \Flagged \Draft)]
* OK [UIDNEXT 63] Predicted next UID
* OK [UIDVALIDITY 1] UID value
* OK [UNSEEN 1] first unseen message
. OK [READ-WRITE] SELECT completed
```

Výpis všech zpráv v adresáři.

```
. FETCH 1:* FLAGS
* 1 FETCH (FLAGS (\Recent))
* 2 FETCH (FLAGS (\Recent))
* 3 FETCH (FLAGS (\Recent))
. OK FETCH completed
```

Přečtení vybrané zprávy.

```
. FETCH 2 body[text]
* 2 FETCH (BODY[TEXT] {39}
Ahoj, toto je testovaci zprava.

 )
. OK FETCH completed
```

Při odhlášení se server rozloučí.

```
. LOGOUT
* BYE dbmail imap server kisses you goodbye
. OK completed
```

Více informací o protokolu IMAP je možné získat v na adrese www.imap.org nebo v [3].

D KONFIGURAČNÍ SOUBOR *MAIN.CF*

```
# See /usr/share/postfix/main.cf.dist for a~commented, more complete version
# Debian specific:  Specifying a~file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = /usr/share/doc/postfix

#------------------SASL------------------
smtpd_recipient_restrictions = reject_unauth_pipelining, permit_mynetworks,
        permit_sasl_authenticated, reject_non_fqdn_recipient,
        reject_unauth_destination, permit


smtpd_sender_restrictions = permit_sasl_authenticated, permit_mynetworks,
        reject_non_fqdn_sender, reject_unknown_sender_domain,
        reject_unauth_pipelining, permit

smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_sasl_path = smtpd
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain =

#Uzitecne pri testovani SASL
#soft_bounce = yes

#------------------TLS-----------------
smtpd_tls_cert_file = /etc/postfix/smtpd.crt
smtpd_tls_key_file = /etc/postfix/smtpd.key
smtpd_use_tls=yes
smtp_tls_security_level = may
smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_loglevel = 1
```

```
smtpd_tls_received_header = yes


#-----Zakladni parametry--------------
myhostname = mail.mojedomena.cz
mydomain = mojedomena.cz
myorigin = mojedomena.cz
mydestination =

alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

relayhost =
#mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mynetwork_style = subnet

#Pro virtualni domeny nutne zakomentovat
#mailbox_command = procmail -a "$EXTENSION"

mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
html_directory = /usr/share/doc/postfix/html

#Nastaveni virtualnich domen
virtual_mailbox_domains = mysql:/etc/postfix/mysql_domains.cf
#Nastaveni virtualnich uzivatelu
virtual_mailbox_maps = mysql:/etc/postfix/mysql_mailbox.cf
#Uloziste - Dbmail
virtual_transport = dbmail-lmtp:[127.0.0.1]:24
#Filtr - Sagator
content_filter = smtp:[127.0.0.1]:27
```

E KONFIGURAČNÍ SOUBOR *MASTER.CF*

```
# Postfix master process configuration file.  For details on the format
# of the file, see the master(5) manual page (command: "man 5 master").
#
# Do not forget to execute "postfix reload" after editing this file.
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       -       -       -       smtpd
submission inet n       -       -       -       -       smtpd
#  -o smtpd_tls_security_level=encrypt
  -o smtpd_tls_auth_only=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,
     reject_unauth_destination,reject
  -o smtpd_sasl_security_options=noanonymous,noplaintext
  -o smtpd_sasl_tls_security_options=noanonymous
#  -o milter_macro_daemon_name=ORIGINATING
smtps     inet  n       -       -       -       -       smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_tls_auth_only=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o smtpd_sasl_security_options=noanonymous,noplaintext
  -o smtpd_sasl_tls_security_options=noanonymous
#  -o milter_macro_daemon_name=ORIGINATING
#628       inet  n       -       -       -       -       qmqpd
pickup    fifo  n       -       -       60      1       pickup
cleanup   unix  n       -       -       -       0       cleanup
qmgr      fifo  n       -       n       300     1       qmgr
#qmgr     fifo  n       -       -       300     1       oqmgr
tlsmgr    unix  -       -       -       1000?   1       tlsmgr
rewrite   unix  -       -       -       -       -       trivial-rewrite
bounce    unix  -       -       -       -       0       bounce
defer     unix  -       -       -       -       0       bounce
trace     unix  -       -       -       -       0       bounce
verify    unix  -       -       -       -       1       verify
flush     unix  n       -       -       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
proxywrite unix -       -       n       -       1       proxymap
smtp      unix  -       -       -       -       -       smtp
# When relaying mail as backup MX, disable fallback_relay to avoid MX loops
relay     unix  -       -       -       -       -       smtp
```

```
        -o smtp_fallback_relay=
#       -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq     unix  n       -       -       -       -       showq
error     unix  -       -       -       -       -       error
retry     unix  -       -       -       -       -       error
discard   unix  -       -       -       -       -       discard
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       -       -       -       lmtp
anvil     unix  -       -       -       -       1       anvil
scache    unix  -       -       -       -       1       scache
#
# ====================================================================
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# Many of the following services use the Postfix pipe(8) delivery
# agent.  See the pipe(8) man page for information about ${recipient}
# and other message envelope options.
# ====================================================================
#
# maildrop. See the Postfix MAILDROP_README file for details.
# Also specify in main.cf: maildrop_destination_recipient_limit=1
#
maildrop  unix  -       n       n       -       -       pipe
  flags=DRhu user=vmail argv=/usr/bin/maildrop -d ${recipient}
#
# See the Postfix UUCP_README file for configuration details.
#
uucp      unix  -       n       n       -       -       pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
#
# Other external delivery methods.
#
ifmail    unix  -       n       n       -       -       pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp     unix  -       n       n       -       -       pipe
  flags=Fq. user=bsmtp argv=/usr/lib/bsmtp/bsmtp -t$nexthop -f$sender $recipient
scalemail-backend unix  -       n       n       -       2       pipe
  flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store
  ${nexthop} ${user} ${extension}
mailman   unix  -       n       n       -       -       pipe
  flags=FR user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py
  ${nexthop} ${user}
```

```
#begin: dbmail
dbmail-lmtp     unix    -       -       n       -       -       lmtp
#end: dbmail

#begin: sagator
127.0.0.1:26    inet n - n - 30 smtpd
 -o content_filter=
 -o myhostname=sagator.b05-524d.kn.vutbr.cz
 -o local_recipient_maps=  -o relay_recipient_maps=
 -o mynetworks=127.0.0.0/8  -o mynetworks_style=host
 -o smtpd_restriction_classes=  -o smtpd_client_restrictions=
 -o smtpd_helo_restrictions=  -o smtpd_sender_restrictions=
 -o smtpd_restriction_classes=  -o smtpd_client_restrictions=
 -o smtpd_helo_restrictions=  -o smtpd_sender_restrictions=
 -o smtpd_recipient_restrictions=permit_mynetworks,reject
#end: sagator
```

F KONFIGURAČNÍ SOUBOR *SAGATOR.CONF*

```
##Dulezite importy
from avlib import *
from scanners import *
from srv import *

## Which mails are considered spam; should be the same as the
## required_hits in spamassassin config.
## Should be declared even in case you don't use the variable in
## your config - it's needed by sagator internal scanners like
## "rewrite".
SPAM_TRESHOLD=4.0

## Rewrites the name of spam to something more useful..
## Used in antispam scanner chains below..

class rewrite_name(rename):
 name='rewrite_name()';
 def scanstream(self,stream):
     level,detected,virlist=match_any.scanstream(self,stream)
     ## Clean messages leave untouched
     if not is_infected(level,detected):
       return level,detected,virlist
     ## These replace variables may be used in new virname definition..
     repl_vars={
       'VNAME': detected,
       'LEVEL': str(level),
       'STARS': '*'*int(level*SPAM_TRESHOLD),
     }
     ## The "detected" is modified, others left unchanged..
     return level,replace_tmpl(self.NEWNAME,repl_vars),virlist


## Debugging level, 0=errors only, 1=return status, init messages,
##   2=smtp server communication, 3=detailed smtp server communication,
##   4=tracebacks, 5=smtp client communication
DEBUG_LEVEL=3

## System settings - Debian standard config
CHROOT='/var/spool/vscan'
LOGFILE=CHROOT+'/var/log/sagator/sagator.log'
USER,GROUP='vscan','vscan'
SMTP_SERVER=('127.0.0.1',26)

## Classes of mails according to infection, may be used in the following
```

```
## scanner configs; unused classes are commented out.
DROP_INFECTED='.'

##Skenery
SCANNERS=[
   log(3, log.SUMMARY_REPORT,
       ##Nejdriv virovy
       status("Virus",
#        report(['petr.smahel@gmail.com'],report.MSG_TMPL,
           quarantine('/tmp/quarantine/%Y%m','',
             drop(DROP_INFECTED,
               buffer2mbox(libclam(db_options=libclam.CL_DB_PHISHING)),
               parsemail(file_type({'exe': 'Executable (Spusitelny soubor)'})),

             )
           )
 #       )

       ),
       ##Potom spamovy
       status("Spam",
           rename('',
               cache('t',
                   spamassassind(['127.0.0.1',783])
               ) * const(0.0)
           ),

           drop('.',
               rename('$STARS',
                   cache('t')>=const(1.75)
               )
           ),
           deliver(
               modify_subject('[%V]',
                   rewrite_name('$VNAME $STARS',
                       cache('t')
                   )
               )
           )
       )
   )
]
SRV=[
 collector('127.0.0.1',28),
 smtpd(SCANNERS,'127.0.0.1',27)
]
```

G OBSAH PŘILOŽENÉHO CD

Název souboru	Obsah
bakalarska_prace.pdf	elektronická verze bakalářské práce
seznam_balicku.pdf	seznam všech nainstalovaných balíčků

Tab. G.1: Seznam souborů na přiloženém CD.