

Firemný firewall s poštovým serverom v operačnom systéme Linux, distribúcia Red Hat.

Corporate firewall and mail server on Linux, Red Hat distribution.

Bc. Michal Hvizdák



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal Hvizdák**
Osobní číslo: **A11715**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Firemní firewall s poštovním serverem pod operačním systémem Linux, distribuce Red Hat**

Zásady pro vypracování:

1. Zpracujte literární řešení na dané téma.
2. Popište vlastnosti a funkce operačního systému Linux.
3. Nakonfigurujte poštovní server s protokoly SMTP, POP3 a IMAP.
4. Vytvořte pomocí serveru bránu z LAN do Internetu s využitím NAT.
5. Na poštovním serveru nakonfigurujte firewall pomocí nástroje Net-filter.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Linux: dokumentační projekt. 4., aktualiz. vyd. Překlad Lubomír Ptáček. Brno: Computer Press, 2007, 1334 s. ISBN 978-80-251-1525-1.**
2. **SMITH, Roderick W. Linux ve světě Windows: průvodce administrátora heterogenních sítí. 1. vyd. Praha: Grada, 2006, 443 s. ISBN 80-247-1470-1.**
3. **DENT, Kyle D. Postfix: kompletní průvodce. 1. vyd. Praha: Grada, 2005, 237 s. ISBN 80-247-1029-3.**
4. **FLICKENGER, Rob. Linux server na maximum: 100 tipů a řešení pro náročné. Brno: CP Books, 2005, 229 s. ISBN 80-251-0586-5.**
5. **KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IP a systémem DNS. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.**
6. **DOSEDĚL, Tomáš. Počítačová bezpečnost a ochrana dat. 1. vyd. Brno: Computer Press, 2004, 190 s. ISBN 80-251-0106-1.**
7. **Správa linuxového serveru [online]. 2009 [cit. 2013-01-28]. ISSN 1214-9608. Dostupné z: <http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru>**
8. **Seriál: Linux prakticky ako server [online]. 2011 [cit. 2013-01-28]. ISSN 1335-4787. Dostupné z: <http://www.itnews.sk/tituly/infoware/2011-05-03/c141036-serial-linux-prakticky-ako-server>**

Vedoucí diplomové práce:

Ing. Miroslav Matýsek, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

22. února 2013


Termín odevzdání diplomové práce:

22. května 2013

Ve Zlíně dne 22. února 2013


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

V práci bol vytvorený e-mailový server založený na operačnom systéme Linux Red Hat. Na začiatku práce je krátko popísaný vznik operačného systému Linux a distribúcie Red Hat, ďalej sú postupne rozoberané všetky potrebné teoretické informácie, ktoré sú využité neskôr pri konfigurácii servera. Na pevných diskoch servera bolo vytvorené softwarové RAID pole a oddiel LVM. Server podporuje protokoly SMTP s autentifikáciou užívateľov, POP3, IMAP a umožňuje pripojenie malej siete do internetu pomocou protokolu NAT. Pre mobilných užívateľov bol nakonfigurovaný webmail. Server vykonáva antivírusovú a antispamovú kontrolu všetkých e-mailov. Na správu firewallu bol vytvorený skript v príkazovom interprete BASH. Činnosť servera sa zaznamenáva na oddiel v rámci LVM pomocou démona *syslogd*. Správu servera uľahčuje grafické webové administračné rozhranie.

Kľúčové slová: Postfix, Dovecot, e-mailový server, firewall, webmail

ABSTRACT

In this thesis was created an e-mail server based on Linux Red Hat. At first it was briefly described the creation of Linux and Red Hat distribution and next all the necessary theoretical information are discussed that are used later in server configuring. On the server hard drives has been created software RAID and LVM partitions. The server supports the SMTP protocol with user authentication, POP3, IMAP protocols and connects the small network to the internet by the NAT protocol. Webmail has been configured for mobile users. The server performs antivirus and antispam control all of the e-mails. The script for the firewall management has been created in the BASH command interpreter. The server activity is logged in the LVM partition by the *syslogd* daemon. Graphical web-based administration interface facilitates the server management.

Keywords: Postfix, Dovecot, e-mail server, firewall, webmail

Touto cestou by som chcel vysloviť poďakovanie môjmu zamestnávateľovi Ústavu informačných a komunikačných technológií na Žilinskej univerzite v Žiline za poskytnutie technických prostriedkov na konfiguráciu e-mailového servera. Ďalej by som rád poďakoval svojej rodine za výborné podmienky, ktoré mi umožnili študovať na vysokej škole a všetkým, ktorí mi pri štúdiu pomáhali a podporovali ma. V neposlednom rade ďakujem vedúcemu mojej práce Ing. Miroslavovi Matýskovi, Ph. D. za odborné vedenie, pomoc a cenné rady, ktoré mi boli nápomocné pri písaní tejto práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	10
I TEORETICKÁ ČASŤ	11
1 UNIX ALEBO LINUX.....	12
1.1 ZAČIATOK SYSTÉMU UNIX A LINUX	12
1.2 LINUX V 21. STOROČÍ.....	14
1.3 RED HAT LINUX	15
2 OPERAČNÝ SYSTÉM LINUX.....	17
2.1 SÚBORY.....	17
2.1.1 Súborový systém	18
2.1.2 Adresárová štruktúra	18
2.1.3 Diskové oddiely.....	19
2.2 RAID	20
2.2.1 Druhy RAID	20
2.3 LVM.....	21
2.4 ÚROVNE BEHU SYSTÉMU	22
2.5 PLÁNOVANÉ SPÚŠŤANIE ÚLOH.....	23
2.6 LOGOVANIE.....	24
2.6.1 Démon syslogd.....	24
2.6.2 Údržba logov	25
3 LINUX AKO FIREWALL	26
3.1 PROGRAM NETFILTER.....	26
3.1.1 Cesta paketu Netfiltrom	27
3.1.2 Príkaz iptables	28
4 E-MAIL	29
4.1 AKO FUNGUJE E-MAIL.....	29
4.2 E-MAILOVÁ SPRÁVA A E-MAILOVÁ ADRESA	30
4.3 E-MAILOVÉ PROTOKOLY	31
4.3.1 SMTP a ESMTP.....	31
4.3.2 POP3	32
4.3.3 IMAP4.....	32
4.4 SMEROVANIE E-MAILOV	33
5 DORUČOVANIE A PRIJÍMANIE POŠTY	34
5.1 DORUČOVANIE POŠTY - SERVER POSTFIX	34
5.1.1 Zložky Postfixu	34
5.1.2 Obecná konfigurácia.....	35
5.1.3 Autentifikácia SMTP	36

5.2	PRIJÍMANIE POŠTY – SERVER DOVECOT	37
5.2.1	POP/IMAP server Dovecot	37
5.2.2	Obecná konfigurácia.....	38
II	PRAKTICKÁ ČASŤ	39
6	ANALÝZA VYTVÁRANÉHO RIEŠENIA	40
6.1	NÁVRH RIEŠENIA	40
7	INŠTALÁCIA A KONFIGURÁCIA SYSTÉMU.....	42
7.1	KONFIGURÁCIA RAID, LVM A ROZDELENIE DISKOV	42
7.1.1	Vytvorenie logického zväzku.....	46
7.1.2	Zväčšenie logického zväzku.....	46
7.1.3	Zmenšenie logického zväzku	47
7.2	NASTAVENIE LOGOVACIEHO DÉMONA	48
7.2.1	Logovanie prihlasovania	48
7.2.2	Logovanie e-mailového systému	49
7.2.3	Reštart logovacej služby.....	49
7.2.4	Konfigurácia rotácie logov	49
7.3	KONFIGURÁCIA SIETE	50
7.3.1	Konfigurácia mena servera.....	50
7.3.2	Konfigurácia sieťového pripojenia.....	50
7.4	VYPNUTIE NEPOTREBNÝCH SLUŽIEB	51
8	INŠTALÁCIA A KONFIGURÁCIA POSTFIX.....	53
8.1	INŠTALÁCIA MTA POSTFIX	53
8.2	POSTFIX AKO PREDVOLENÝ MTA.....	54
8.3	KONFIGURÁCIA POSTFIX	54
8.4	KONFIGURÁCIA AUTENTIFIKÁCIE SMTP.....	55
8.4.1	Inštalácia Cyrus SASL	55
8.4.2	Konfiguračný súbor pre Postfix	56
8.4.3	Príprava Postfix na použitie SMTP AUTH.....	57
8.4.4	Databáza sasldb2	58
8.4.5	Test autentifikácie	59
8.5	VYTVORENIE E-MAILOVÉHO ÚČTU.....	60
9	INŠTALÁCIA ANTIVÍRUS A ANTISPAM	61
9.1	PRIDANIE REPOZITÁROV EPEL.....	61
9.2	INŠTALÁCIA CLAMAV.....	61
9.3	INŠTALÁCIA MAILSCANNER A SPAMASSASSIN.....	62
9.4	KONFIGURÁCIA MAILSCANNER A SPAMASSASSIN	63
9.5	TEST NASTAVENIA	65
10	KONFIGURÁCIA PRÍJÍMU E-MAILOV PRE DOMÉNU.....	66
11	INŠTALÁCIA A KONFIGURÁCIA DOVECOT	68

11.1	INŠTALÁCIA DOVECOT	68
11.2	KONFIGURÁCIA DOVECOT	68
12	INŠTALÁCIA A KONFIGURÁCIA WEBMAIL.....	70
12.1	INŠTALÁCIA SERVERA APACHE A PHP	70
12.2	INŠTALÁCIA A KONFIGURÁCIA SQUIRRELMAIL.....	70
12.3	INŠTALÁCIA DOPLNKOV SQUIRRELMAIL	73
12.3.1	Inštalácia Retrieve User Data plugin.....	73
12.3.2	Inštalácia Autocomplete plugin.....	73
13	INŠTALÁCIA A KONFIGURÁCIA FIREWALL	74
13.1	INŠTALÁCIA FIREWALLU	74
13.2	ZÁKLADNÉ NASTAVENIE SYSTÉMU	74
13.3	SKRIPT FIREWALLU	74
14	NÁSTROJE NA GRAFICKÚ SPRÁVU SERVERA.....	76
14.1	INŠTALÁCIA PROGRAMU WEBMIN	76
14.2	ZÁKLADNÉ NASTAVENIA	77
	ZÁVER	78
	CONCLUSION	80
	ZOZNAM POUŽITEJ LITERATÚRY	82
	ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK	84
	ZOZNAM OBRÁZKOV	87
	ZOZNAM TABULIEK	88
	ZOZNAM PRÍLOH.....	89

ÚVOD

Spôsoby, ktoré využívame na komunikáciu sa neustále vyvíjajú a menia. Ľudská komunikácia bola kedysi silno obmedzená len na komunikáciu typu tvárou v tvár. V dnešnej dobe, je však spôsobov, ktoré využívame na komunikáciu s ostatnými ľuďmi okolo nás mnoho a výraznou zložkou našej komunikácie je práve komunikácia prostredníctvom elektronických dátových sietí. Spôsobov komunikácie a posielania správ je mnoho, či už ide o nejakú formu rýchlych správ posielaných prostredníctvom dátovej siete v reálnom čase, využívanie systémov fungujúcich na základe internetovej telefónie akým je Skype alebo obyčajné telefonovanie ako ho všetci poznáme. Väčšina ľudí si však stále bude pod pojmom elektronická komunikácia predstavovať určitú formu elektronickej pošty.

Elektronická pošta (e-mail) je jednou z najdôležitejších služieb dnešného sveta, je využívaná celosvetovo a pozná ju snáď každý. Denne sú na svete posielané miliardy e-mailov, čo kladie veľké požiadavky na ich strojové spracúvanie, triedenie a samozrejme aj odovzdávanie tým správnym užívateľom, ktorým sú adresované. Poštové servery plniace tieto funkcie sú po celom svete a spracúvajú správy bez tušenia ľudí, ktorí ich posielajú.

Problematika konfigurácie a správy takéhoto e-mailového servera, ktorý plní v sieti viacero úloh ma veľmi zaujíma a bola hlavným dôvodom prečo som si vybral práve túto tému. Táto práca bola venovaná predovšetkým vytvoreniu poštového servera, ktorý umožní užívateľom odosielať a prijímať e-mailové správy vo svojom mene, triediť ich do rôznych priečinkov a spracovávať ich. Je veľmi žiadané a moderné, aby mali užívatelia prístup k svojim dátam prakticky kdekoľvek. Možnosť pristupovať napríklad k pracovným e-mailom by mala byť samozrejmosťou aj z domáceho prostredia, preto je nevyhnutné použitie webmailu. Je zrejmé, že server musí byť pripojený do verejnej siete Internet, aby mohol poskytovať svoje služby, z čoho plynú rôzne riziká, preto musíme dbať na jeho bezpečnosť, či už fyzickú, alebo softvérovú a samozrejme aj na bezpečnosť údajov užívateľov, ktorý daný server používajú. Veľmi dôležitá je spoľahlivosť takéhoto servera. Každý z nás už určite zažil situáciu, keď prišiel o nejaké osobné dáta, strata e-mailov, nie je takisto nič príjemné. Všetky IP adresy, ktoré boli použité v práci sú vymyslené rovnako aj firma Viphome s.r.o.

I. TEORETICKÁ ČASŤ

1 UNIX ALEBO LINUX

Linux je v súčasnosti slovo, ktoré je oveľa známejšie ako tomu bolo v minulosti. Niektorí ľudia nevedia, čo si majú pod týmto pojmom predstaviť, nevedia, že Linux je operačný systém, nevedia, že ho používajú v mobile, v tablete, v satelitnom prijímači, či dokonca v hodinkách. Jeho použitie je veľmi široké, hodí sa na veľké spektrum vecí, je spoľahlivý, otvorený a je zadarmo.

1.1 Začiatok systému UNIX a Linux

Keď sa presunieme do minulosti asi štyridsať rokov späť, musíme si predstaviť počítače veľké ako domy, či celé budovy. Veľký problém týchto strojov bola ich veľkosť, no nemenej väčším problémom bol samotný operačný systém, ktorý tieto počítače ovládal. Nebol nijako štandardizovaný a každý počítač používal nejaký iný, ktorý samozrejme fungoval len na ňom. Program, ktorý fungoval na jednom počítači sa pri použití na inom musel kompletne prepísať a naprogramovať tak, aby fungoval na inom operačnom systéme.

„Technológia tej doby nebola ešte dostatočne vyspelá, takže bolo nutné sa s veľkosťou počítačov zmieriť ešte na celých desať rokov. Medzitým, v roku 1969, začal tím výskumníkov v Bellových laboratóriách pracovať na riešení softwarového problému. Vyvinuli nový operačný systém, ktorý bol:

1. Jednoduchý a elegantný.
2. Napísaný v jazyku C, nie v Assembleri.
3. Schopný recyklácie kódu.

Vývojári v Bellových laboratóriách tento projekt pomenovali UNIX [11].“

Za ľudí, ktorí vymysleli UNIX sa považujú páni Ken Thompson a Dennis Ritchie, títo ostali pracovať vo firme Bell Labs po neúspešnom projekte Multics (Multiplexed Information and Computing Service) asi najslávnejšom softvérovom systéme, ktorý bol úplne nanič, pretože nefungoval [7].

Dovtedy dodávané komerčné operačné systémy boli napísané vždy v jazyku, ktorý bol konkrétne vyvinutý pre daný systém. Oproti tomu UNIX potreboval len malú časť špeciálneho kódu pre hardvér, na ktorom mal byť spustený, čomu dnes hovoríme jadro.

Samotný operačný systém a jeho funkcie boli naprogramované v jazyku pôvodne vyvinutom len pre tento účel – v jazyku C.

V osemdesiatych rokoch už domáce počítače neboli výnimkou, veľa ľudí ich vlastnilo a v tej dobe už existovalo pre platformu PC niekoľko variant UNIXU. Nevýhodou však bolo, že žiadna z nich nebola ľahko dostupná a väčšina z nich bola veľmi pomalá, čo viedlo užívateľov k používaniu MS-DOS (Microsoft Disc Operating System) alebo Windows 3.1.

Na začiatku 90. rokov boli už osobné počítače dostatočne výkonné a v tom čase mladý fínsky študent a výborný programátor Linus Torvalds skombinoval všetko čo sa naučil ako študent počítačových vied na univerzite v Helsinkách a naprogramoval jadro vlastného operačného systému, ktoré neskôr prinieslo na svet Linux [11].

Veľkým problémom pri jeho tvorbe bola uzavretosť zdrojových kódov, čo znamená, že Linus nemohol meniť existujúce prostredie operačného systému, pretože vlastnil len jeho binárnu verziu.

„Riešenie sa naskytlo vo forme projektu GNU GPL (GNU General Public License). Cieľom tohto projektu bolo vytvoriť voľne šíriteľný klon UNIXU, ktorý by bol k dispozícii vrátane zdrojových kódov úplne každému [10].“

Tak v roku 1991 vznikla prvá verzia dnešného Linuxu. Za dva roky mal Linux 12 000 užívateľov. Projekt obľúbený medzi počítačovými nadšencami sa trvalo rozrastal, pričom sa stále držal štandardu POSIX (Portable Operating System Interface). V roku 1993 Linus dokončil prvú verziu nového unixového klonu.

„Aby sa do vývoja mohlo zapojiť čo najviac ľudí na svete, a doviest' tak celý projekt ku skutočnej profesionalite, musel svoj výtvor umiestniť na internet a upozorniť naň. A tak sa na jednom z FTP (File Transfer Protocol) serverov, kam môže ktokoľvek umiestniť svoj výtvor, a dať ho ďalej k dispozícii ostatným objavil Linux [10].“

Ako by si mnohí mysleli, názov nepochádza priamo od tvorca, ale od administrátora tohto servera, ktorý Linusov projekt nazval Linux – **Linus**ov Minix.

Tým začala nová éra UNIXU a v priebehu niekoľko rokov boli doplnené všetky funkcie unixových systémov, takže výsledkom je dnešný vyzretý operačný systém Linux, ktorý je

plnohodnotným klonom systému UNIX, použiteľným ako na pracovných staniciach, tak na špičkových serveroch.

1.2 Linux v 21. storočí

Linux sa ako všetky operačné systémy postupne vyvíjal, pričom prinášal stále nové a nové funkcie. V súčasnej dobe ho nájdeme aj na trhu s osobnými počítačmi a jeho podiel tvorí asi dve percentá. S príchodom modernej doby a rôznych nových technológií sa však jeho úloha na rozdiel od minulosti podstatne mení. V minulosti ho na svojom počítači chcelo mať mnoho ľudí hlavne kvôli možnosti testovať v ňom nové hardvérové komponenty, vďaka čomu vznikali voľne dostupné ovládače pre širokú množinu hardvéru a aj pre rôznych „exotický“ hardvér. Dnes má Linux na svojom počítači skoro každý z nás bez toho, aby o tom vedel. Počítačom nemyslíme skutočný stolový počítač alebo notebook, ale najnovší výdobytok modernej techniky, ktorý nazývame smartphony či tablety. Bez Linuxu sa nezaobíde ani svet internetu.

„Čo sa týka použitia na serveroch, má Linux povest' stabilnej a spoľahlivej platformy, na ktorej bežia databázové služby takých spoločností, ako je Amazon, známe elektronické knihkupectvo, americká pošta, nemecká armáda a ďalšie. Veľmi obľúbený je Linux u poskytovateľov internetového prístupu a služieb, kde sa používa ako firewall, proxy server alebo webový server. Počítač s Linuxom nájdete i u každého správcu niektorého Unixového systému, ktorý ho používa ako pohodlnú administratívnu stanicu. Clustery linuxových počítačov sa podieľali na vzniku filmov ako Titanic alebo Shrek. Na poštách slúžia ako centrály riadiace smerovanie zásielok, veľké vyhľadávacie stroje pomocou nich prehľadávajú Internet [11].“

Samozrejmosťou je, že medzi úlohy, ktoré dokáže zvládnuť môžeme zaradiť aj úlohy poštového servera. Dôležitým rozhodnutím pri konfigurácii je výber vhodnej distribúcie. V tejto práci bola použitá distribúcia Red Hat, ktorá je primárne určená najmä pre firemné serverové použitie.

1.3 Red Hat Linux

RHEL (Red Hat Enterprise Linux) je produktom americkej spoločnosti Red Hat, Inc., ktorá sídli v meste Raleigh, v Severnej Californii. Spoločnosť bola založená v roku 1990 spojením dvoch spoločností majiteľov Marca Ewinga a Boba Younga.

V roku 1993 Bob Young vytvoril spoločnosť ACC Corporation (Antiques and Collectables of Connecticut), ktorej skratka bola odvodená od rovnomenného názvu malého podniku jeho ženy. Približne v tom istom čase Marc Ewing vytvoril vlastnú Linuxovú distribúciu, ktorú nazval Red Hat Linux podľa ním noseného a obľúbeného červeného klobúka, ktorý sa neskôr stal logom firmy (Obr. 1). V roku 1995 spoločnosť ACC získala Red Hat Linux a prijala meno Red Hat, Inc., spoločnosť vstúpila na trh a jej majitelia zbohatli. Prvé verzie tohto systému boli zákazníkom predávané na disketách prípadne CD, hlavne kvôli malej dostupnosti pripojenia do internetu. Firma Red Hat v tom čase poskytovala podporu pre užívateľov svojho systému, avšak len prostredníctvom e-mailu, čo vlastne redukovalo počet požiadaviek na riešenie nejakého problému, keďže väčšina užívateľov mala len jeden počítač, v prípade poruchy systému alebo akejkoľvek inej hardwarovej poruchy firmu kontaktovať e-mailom nemohli [1].

RHEL je komerčným systémom, ktorý je zameraný hlavne na inštalácie vo firmách a na použitie na kritické úlohy. Zdrojové kódy RHEL sú voľne k dispozícii, takisto aj kódy open-source programov a aktualizácií, vďaka čomu boli vytvorené rôzne iné distribúcie Linuxu, ktoré sú klonom tohto systému. Najznámejšie z nich sú:

- CentOS (Community Enterprise Operating System),
- Oracle Linux,
- Scientific Linux.

V roku 2003 zmenila firma Red Hat, Inc. svoju obchodnú stratégiu a rozhodla sa vyvíjať len Red Hat Enterprise Linux, ktorý je určený predovšetkým pre podnikovú sféru. Reakciou na tento krok bol vznik projektu Fedora, o ktorý sa stará komunita vývojárov, ktorí sú čiastočne sponzorovaní práve firmou Red Hat. Fedora je vyvíjaná s dôrazom na používateľský komfort a na použitie na domácich počítačoch avšak nevylučuje sa ani použitie na serveroch.



Obr. 1 – Logo distribúcie Red Hat [4].

2 OPERAČNÝ SYSTÉM LINUX

Každý operačný systém sa skladá z viacerých častí a modulov, väčšinou ho však tvorí jadro systému a počítačový software. Úlohou operačného systému je poskytovať podporu pre beh počítačových programov. Linux sa považuje za stredne veľký operačný systém, ktorého filozofiou je deliť úlohy do malých častí a teda do relatívne jednoduchých programov. Najdôležitejšou časťou každého operačného systému, pričom Linux nie je výnimkou je jadro. Jadro býva často nesprávne považované za celý operačný systém, ako také však toho vykonáva veľmi málo. Jeho najdôležitejšou činnosťou je poskytovať svoje služby iným nástrojom, pomocou ktorých sú realizované všetky ostatné služby v systéme ako sú časti na riadenie procesov, správa pamäti, ovládače súborových systémov, správa siete a podobne.

2.1 Súbory

„V unixovom systéme je všetko súbor. Ak niečo súbor nie je, je to proces [11].“

Linux nerozlišuje, medzi adresármi a súbormi. Aj adresár je len špeciálnym typom súboru, ktorý však obsahuje názvy iných súborov, ktorými môžu byť programy, služby, textové súbory, obrázky a podobne. Typov súborov je v Linuxe niekoľko a sú uvedené v nasledujúcej tabuľke (Tab. 1).

Tab. 1 – Typy súborov v Linuxe [11].

Znak	Význam
-	Normálny súbor
d	Adresár
l	Odkaz
c	Znakové zariadenie
s	Soket
p	Pomenovaná rúra
b	Blokové zariadenie

Všetky súbory sú uložené pomocou nejakého súborového systému a v adresárovom strome.

2.1.1 Súborový systém

Súborový systém je spôsob akým sú súbory pomenované, kde sa logicky umiestňujú na úložné médium pri ukladaní, aby mohli byť k dispozícii pri čítaní. Je to prostriedok organizácie dát na diskovom oddieli. Ďalšou nemenej dôležitou funkciou je správa voľného miesta na úložnom médiu. V Linuxe rozlišujeme viacero súborových systémov, z nich každý má svoje výhody a nevýhody a možnosti využitia. V každom súborovom systéme nájdeme minimálne jeden adresár, ktorý sa nazýva hlavný alebo koreňový adresár. V Linuxe ho označujeme znakom „/“. Najznámejšie a najpoužívanéjšie súborové systémy Linuxu sú:

- **Ext2 – Second Extended Filesystem** – starší súborový systém, ktorý je veľmi stabilný, môže byť konvertovaný na ext3.
- **Ext3 – Third Extended Filesystem** – je v podstate systém ext2 s pridanou podporou žurnálovania a je spätne kompatibilný s ext2.
- **Ext4 – Fourth Extended Filesystem** – je novší systém súborov spätne kompatibilný s ext2 a ext3. Jeho výhodou je podpora kapacity až do 1 exabyte a súborov veľkých viac ako 16 terabytov [5].
- **XFS – Extended File System** – žurnálovací systém súborov. Poskytuje vysokú priepustnosť pri práci s veľkými súbormi, problémom je veľa malých súborov.
- **Swap** – systém súborov, ktorý sa používa pre swapovacie diskové oddiely.

Pri konfigurácii servera bol využitý súborový systém ext4.

2.1.2 Adresárová štruktúra

Organizácia adresárovej štruktúry Linuxu je úplne odlišná od systému adresárov napr. operačného systému Windows. Štruktúra adresárov Linuxu sa môže zdať na prvý pohľad nelogická a ťažká na pochopenie a môže byť ťažké sa v nej orientovať a hľadať jednotlivé programy, ich konfiguračné súbory a podobne. V drvivej väčšine prípadov bude adresárová štruktúra linuxového systému po použití príkazu *ls* nasledovná:

```
[michal@pluto /]$ ls
```

```
bin dev home lib64 media mnt opt root selinux sys usr  
boot etc lib lost+found misc net proc sbin srv tmp var
```

Vo väčšine Linuxových distribúcií bude výstup príkazu *ls* podobný vďaka norme FSSTND (File System Standard), ktorá sa snaží zaviesť určité konvencie do organizácie adresárového stromu. Adresárová štruktúra bola navrhnutá spôsobom, aby mohla fungovať aj v počítačovej sieti, kedy jej časti môžu byť zdieľané s inými počítačmi napríklad pomocou protokolu NFS (Network File System). Linux dovoľuje svoju adresárovú štruktúru akokoľvek rozdeliť a teda môže byť uložená na viacerých fyzických diskoch alebo na samostatných diskových oddieloch.

2.1.3 Diskové oddiely

Každý pevný disk môže byť rozdelený na viacero diskových oblastí – tzv. partícií. Na začiatku každého pevného disku (prvý sektor, prvá stopa, prvá vrstva disku) sa nachádza tabuľka, ktorá obsahuje údaje o vytvorených partíciách na disku, ktorých počet je z historických dôvodov obmedzený na číslo 4. Počet primárnych partícií väčšinou nestačí, preto je možné definovať jednu z primárnych partícií ako rozšírenú a táto rozšírená partícia môže obsahovať ďalšie partície. Jednotlivé partície sú typom špeciálneho súboru a sú uložené v adresári */dev/*. Viacero partícií môžeme používať aj v rámci jedného operačného systému. Pri inštalácii servera by sme k tomuto kroku mali pristúpiť, pretože zvyšuje bezpečnosť servera hneď z niekoľkých dôvodov:

- bezpečnosť uložených údajov – v prípade softvérového poškodenia partície budú poškodené len dáta na konkrétnej partícii,
- bezpečnosť servera – pre súborové systémy na jednotlivých partíciách môžeme nastaviť odlišné parametre (kvóty, práva, iba na čítanie), čím zvýšime bezpečnosť uložených údajov,
- ochrana pred preplnením – oddelením užívateľských údajov od systémových môžeme zamedziť pádu systému v prípade, že bude vyčerpaná kapacita partície.

V operačnom systéme Linux existujú dva hlavné typy diskových oddielov:

- Dátový oddiel – bežný diskový oddiel určený na ukladanie dát.
- Odkladací oddiel – rozšírenie fyzickej pamäte počítača [11].

Pri inštalácii servera sme zobrali do úvahy dôvody uvedené v tejto časti a bolo navrhnuté rozdelenie partícií tak ako je uvedené v tabuľke (Tab. 2).

Tab. 2 – Odporúčaná konfigurácia diskových oddielov servera.

Pripájací bod	Popis
	SWAP
/	Koreňový súborový systém
/boot	Umiestnenie jadra systému
/home	Domáce adresáre používateľov systému
/var	Súbory meniace sa počas behu systému
/var/log	Logovacie záznamy
/var/spool/mail	Úložisko e-mailov
/tmp	Dočasné súbory – verejný adresár

2.2 RAID

Skratka RAID (Redundant Array of Independent Disks) sa prvý krát objavila v roku 1987, kedy Kalifornská univerzita v Berkeley publikovala článok o využití redundantných polí lacných diskov. V článku bolo popísaných niekoľko typov diskových polí, ktoré sa označovali skratkou RAID. Základnou myšlienkou RAID je spojenie niekoľko nezávislých diskov do jedného poľa, ktoré bude výkonnejšie ako jeden veľký a drahý disk a systému sa bude toto pole javiť ako jedna logická jednotka [11].

Technické riešenie RAID je dvojaké:

- softwarový RAID – o spojenie diskov sa stará jadro operačného systému,
- hardwarový RAID – o spojenie diskov sa stará špecializovaný radič.

2.2.1 Druhy RAID

Typov RAID je mnoho, niektoré sa však prakticky vôbec nepoužívajú, alebo sa používajú len vo veľmi špecializovaných systémoch. Zameriame sa teda len na tie najzaujímavejšie, ktoré sú:

- **Lineárny mód** – dva alebo viac diskov sú spojené do jedného fyzického zariadenia, sú spojené lineárne za seba, takže zápis na RAID prebieha lineárne – najskôr sa zapisuje na disk 0 a po zaplnení na disk 1 atď. [11].

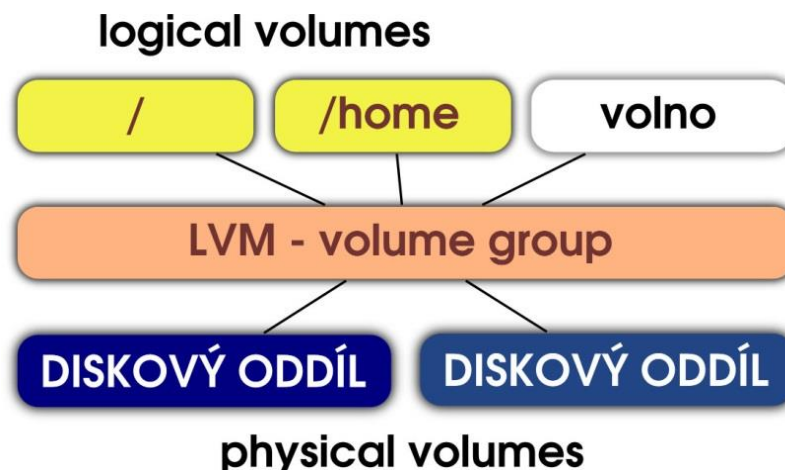
- **RAID 0** – dáta sa rozmiestňujú striedavo po všetkých diskoch z poľa, ktoré by mali mať rovnakú veľkosť, nie je to však podmienka. Napríklad sa zapisuje po 4 KB blokoch postupne od prvého po posledný disk. Strata jedného disku znamená stratu všetkých dát [15].
- **RAID 1** – dáta sa zrkadlia na všetky disky z poľa, pričom kapacita je rovná najmenšiemu disku. Všetky disky musia mať rovnakú kapacitu. Pole dokáže ochrániť pred stratou $n-1$ diskov (n – počet diskov v poli) [15].
- **RAID 5** – minimom pri tomto type RAID sú tri disky. Dáta sú rozdelené do blokov a sú zapisované striedavo na všetky disky s paritou na každom disku. Pole dokáže ochrániť pred stratou $n-1$ diskov [15].
- **RAID 6** – minimom pri tomto type RAID sú štyri disky. Dáta sú delené do blokov ako v prípade RAID 5, používa však dva bloky s paritou na každom disku [15].
- **RAID 10** – tento RAID je kombináciou dvoch popisovaných v poradí RAID 1 a RAID 0. Najskôr sa vytvorí RAID 1 a nad týmto poľom sa vytvorí RAID 0.

2.3 LVM

LVM (Logical Volume Manager) je skratkou pre systém, ktorý slúži na správu logických oddielov – diskových oddielov. Je to mechanizmus mapovania jedného blokového zariadenia na druhé.

LVM má viacero výhod, ktoré sa s klasickými diskovými oddielmi robiť nedajú. Napríklad je možné veľkosť konkrétneho diskového oddielu za chodu serveru zväčšiť alebo zmenšiť podľa potreby. Ďalšou výhodou je možnosť vytvárať nové a rušiť staré oddiely za chodu servera. Nespornou výhodou je aj možnosť robiť tzv. snapshoty (uloženie stavu systému v určitom časovom okamihu) [14].

LVM si môžeme predstaviť ako vrstvu, do ktorej na jednej strane vstupujú fyzické blokové zariadenia a na druhej strane nám dovoľuje vytvoriť z nich logické zväzky, na ktorých môžeme vytvoriť ľubovoľné súborové systémy a manipulovať s nimi za behu systému tak ako to vidíme na obrázku (Obr. 2), čo má veľký zmysel hlavne pri použití na serveroch.



Obr. 2 – Princíp činnosti LVM [14].

2.4 Úrovne behu systému

Úrovne behu systému reprezentujú stav systému v akom má byť spustený, teda ktoré služby majú byť spustené a ktoré naopak majú byť zastavené. Týchto úrovní je šesť, ak nepočítame úroveň číslo nula, ktorá znamená zastavenie systému. Označujú sa číselne tak ako vidíme v tabuľke (Tab. 3).

Služby, ktoré sa majú pri jednotlivých úrovniach behu systému spúšťať a zastavovať sú umiestnené v adresári */etc/rcN.d/* (môže sa líšiť podľa distribúcie), kde **N** je číslo úrovne behu. V týchto adresároch sú umiestnené odkazy na štartovacie skripty jednotlivých služieb, ktoré odkazujú do adresára */etc/init.d/*. Jednotlivé odkazy začínajú písmenom **S** (služba sa bude spúšťať) alebo **K** (služba bude zastavená, ak je spustená). Nasleduje číselný údaj, ktorý určuje poradie spúšťania a zastavovania jednotlivých služieb.

To v akej úrovni bude systém spustený určuje súbor */etc/inittab*. Zmenou tohto súboru môžeme systém zavádzať do iného režimu, prípadne môžeme medzi úrovňami behu prepínať priamo z príkazového riadka pomocou príkazu *init úroveň_behu*:

```
[root@pluto /]# init 5
```

Tab. 3 – Úrovně behu systému Linux [11].

Úroveň behu	Popis
0	Zastavenie systému
1	Režim jedného užívateľa
2	Lokálny viac užívateľský režim so sieťou, ale bez sieťových služieb
3	Plne viac užívateľský režim so sieťou
4	Nepoužíva sa
5	Plne viac užívateľský režim so sieťou a GUI
6	Reštart systému

2.5 Plánované spúšťanie úloh

Démon *cron* zaisťuje pravidelné spúšťanie programov, ktoré číta zo záznamov v užívateľských tabuľkách nazývaných aj „crontaby“. Rozlišujeme systémový crontab, na zmenu ktorého má práva len užívateľ *root* a užívateľské crontaby. Démon *cron* hľadá v adresári */var/spool/cron/* súbory rovnomenné užívateľským účtom systému a prehľadáva adresár */etc/cron.d/*, z ktorého načítava súbor */etc/crontab* [11].

Editáciu užívateľského crontabu môžeme začať príkazom:

```
[michal@pluto ~]$ crontab -e
```

Tento príkaz otvorí prednastavený systémový editor, v ktorom môžeme upraviť náš crontab, ktorý bude na začiatku s najväčšou pravdepodobnosťou prázdny. Postupne budeme teda zapisovať príkazy, ktoré chceme v špecifikovaný čas vykonať.

Jednotlivé príkazy sa zapisujú na samostatné riadky. Prvý údaj definuje minúty (0 – 59), druhý údaj hodiny (0 – 23), tretí údaj deň v mesiaci (1 – 31), štvrtý údaj mesiac (1 -12) a piaty deň v týždni (0 – 7, pričom 0 aj 7 znamená nedeľa). Jednotlivé stĺpce sú od seba oddelené minimálne jednou medzerou a hodnotou stĺpca môže byť číslo, zoznam čísel, rozsah čísel, alebo hviezdička (predstavuje všetky platné hodnoty) [10].

2.6 Logovanie

Prvé, čo urobí zdatný útočník po úspešnom preniknutí do servera je, že vypne logovaciu službu, hlavne kvôli tomu, aby nemohla byť jeho činnosť vykonaná v systéme neskôr analyzovaná.

Log znamená zápis, záznam alebo protokol. Logovanie je činnosť nejakého procesu, prípadne programu – démona, ktorý zaznamenáva určité informácie do súborov s cieľom ich neskoršej analýzy administrátorom [12].

Niektoré procesy v operačnom systéme Linux využívajú svojich vlastných logovacích démonov a informácie zaznamenávajú do vlastných logovacích súborov. Niekedy je však lepšie používať tzv. centralizované logovanie, kedy sa o logovanie väčšiny udalostí v systéme stará jediný proces, ktorý je určený len na túto činnosť. V systéme Linux je asi najpoužívanejšou službou centralizovaného logovania démon *syslogd*.

2.6.1 Démon syslogd

Konfiguračný súbor démona *syslogd* sa väčšinou nachádza v adresári */etc/* a má meno *syslog.conf*, v prípade systému Red Hat je to súbor *rsyslog.conf*. Konfigurácia pravidiel je na samostatných riadkoch v tvare:

- *facility.priority* *akcia*.

Facility (z angl. zariadenie, prostriedok) v našom prípade predstavuje zdroj logovanej správy alebo jej typ. Priority (z angl. prednosť, priorita) označuje dôležitosť logovacej správy. So zvyšujúcou sa dôležitosťou sa znižuje počet vytvorených správ logovacím démonom. Akcia určuje spôsob manipulácie s jednotlivými správami (napr. presmerovanie do konkrétneho súboru, presmerovanie na iný server a pod.) [12].

V nasledujúcej tabuľke (Tab. 4) vidíme možno kombinácie hodnôt jednotlivých nastavení.

Tab. 4 – Příklad hodnôt parametrov „facility“ a „priority“.

Facility	Priority
auth – autentifikačné správy	debug – veľmi podrobné informácie
cron – správy démona cron	info – všeobecné podrobné informácie
daemon – správy démonov	notice – informatívne hlásenia
kern – správy jadra	warn – varovné hlásenia
lpr – správy tlačového portu	err – chybové hlásenia
mail – správy e-mailového systému	crit – hlásenia o kritických chybách
syslog – správy syslogd	alert – výstražné hlásenia
user – správy užívateľských procesov	emerg – panické hlásenia

2.6.2 Údržba logov

Informácii, ktoré sa zapisujú do logovacích súborov je v systéme Linux veľmi veľa. Tieto súbory môžu počas činnosti servera, ktorý je nepretržite v prevádzke narásť do veľkých rozmerov a zaplniť tak aj úložné médium s vysokou kapacitou. Preto pristupujeme k archivácii a mazaniu starých logovacích súborov.

Pri archivácii sa vytvorenému archívu z pôvodného súboru prideli číslo 1 a pôvodný súbor sa vyprázdni. Znovu po stanovenom čase sa archívu s číslom 1 prideli číslo 2, súčasný logovací súbor sa archivuje, prideli sa mu číslo 1 a pôvodný súbor sa vyprázdni. Tak sa postupuje až po číslo 4, ktoré je možné zmeniť. Najstarší súbor s číslom 4 sa neskôr vymaže a takto sa postupuje stále dookola. Tejto technike hovoríme rotácia logov [16].

O rotáciu logov sa stará program *logrotate*, ktorý sa spúšťa prostredníctvom démona *crond*. V adresári */etc/cron.daily/* nájdeme skript, ktorý spúšťa program *logrotate*. Konfiguračný súbor tohto programu nájdeme v adresári */etc/* pod menom *logrotate.conf* a pomocou neho je nakonfigurovaná samotná rotácia logov, čísla pridelované jednotlivým archívom, periodicita rotácie a podobne.

3 LINUX AKO FIREWALL

„Firewall je sada opatření (hardwarových, softwarových či personálních), které mají za cíl přepojit dvě alebo více sítí s různou úrovní důveryhodnosti tak, že sníží (vopred definované) riziká vyplývající pre chránené siete z tohto prepojenia [3].“

Firewall môže byť vyhradené zariadenie, ktoré je špecializované a vytvorené len na túto funkciu, veľmi často je to však softwarové vybavenie zaisťujúce príslušnú funkčnosť, ktoré inštalujeme a konfiguruujeme na nejakom serveri. Vylúčené nie sú ani kombinácie jednotlivých možností. Základné technológie sú:

- aplikačný proxy server – napr. server *squid* pracujúci na aplikačnej vrstve,
- paketový filter – analyzuje IP adresy (sieťová vrstva) a porty (transportná vrstva).

V tejto práci sa budeme zaoberať softwarovým firewallom, konkrétne programom Netfilter, ktorý je známy aj pod pojmom Iptables.

3.1 Program Netfilter

Netfilter je paketový filter, t. j. nie je to priamo firewall. Pracuje na úrovni jadra systému Linux a umožňuje filtrovať packety na základe mnoho kritérií. Samotný Netfilter tvorí skupina tabuliek s pravidlami, pomocou ktorých jadro riadi tok a filtrovanie packetov. Umožňuje nakonfigurovať radu vecí od prekladu adres – NAT (Network Address Translation), prekladu portov – PAT (Port Address Translation) a rôzne iné. Ďalšou možnosťou je však nastaviť ho tak, aby plnil úlohu firewallu.

Firewall sa rozhoduje na základe pravidiel, ktoré sa skladajú z podmienky a akcie (v jednom pravidle môže byť viac podmienok aj akcií). Podmienky sa stanovujú pre jednotlivé údaje (porty, IP adresy), ktoré môžeme vyčítať z dátového toku. Údaje získané z dátového toku sa porovnávajú s pravidlami, ktoré sa zapisujú do tzv. chains – reťazí. Pri porovnávaní sa postupuje od vrchu nadol, až kým packet nejakej podmienke nevyhoví. Reťaze sú uložené v tabuľkách a môžu byť vstavané alebo definované užívateľom. V prípade, že packet vyhoví nejakému pravidlu vykonáva sa niektorá z možných akcií, v rámci ktorej môže byť packet zahodený (DROP), odmietnutý (REJECT), prijatý (ACCEPT) alebo predaný inej reťazi [17].

Ak packet nevyhovuje žiadnemu pravidlu v danej reťazi a jedná sa o jednu zo vstavaných reťazí sú k dispozícii dve možnosti – packet prijať (ACCEPT) alebo odmietnuť (DROP) – tzv. implicitná akcia. Ak sa jedná o užívateľsky definovaný reťaz je packet vrátený tam, odkiaľ bol poslaný (RETURN) [18].

Netfilter obsahuje tri typy tabuliek, z ktorých každá obsahuje iný typ vstavaných reťazí tak ako vidíme v tabuľke (Tab. 5):

Tab. 5 – Tabuľky a reťaze programu Netfilter.

Vstavané reťaze	Tabuľka		
	FILTER	NAT	MANGLE
INPUT	Obsahuje	Neobsahuje	Obsahuje
FORWARD	Obsahuje	Neobsahuje	Obsahuje
OUTPUT	Obsahuje	Obsahuje	Obsahuje
PREROUTING	Neobsahuje	Obsahuje	Obsahuje
POSTROUTING	Neobsahuje	Obsahuje	Obsahuje

Tabuľka **filter** je predvolenou tabuľkou pre všetky zadávané pravidlá bez uvedenia tabuľky. Obsahuje filtrovacie pravidlá a má tri vstavané reťaze (INPUT, FORWARD, OUTPUT). Tabuľka **nat** sa používa hlavne na preklad zdrojovej alebo cieľovej adresy v príslušných poliach v rámci IP packetu a má taktiež tri vstavané reťaze (OUTPUT, PREROUTING, POSTROUTING). Tabuľka **mangle** sa používa na špecifické úpravy packetov a pravidlá v nej môžu byť použité napríklad na úpravu hodnoty TTL (Time to Live) alebo zmenu TOS (Type of Service). Obsahuje všetky vyššie spomínané reťaze.

3.1.1 Cesta paketu Netfiltrom

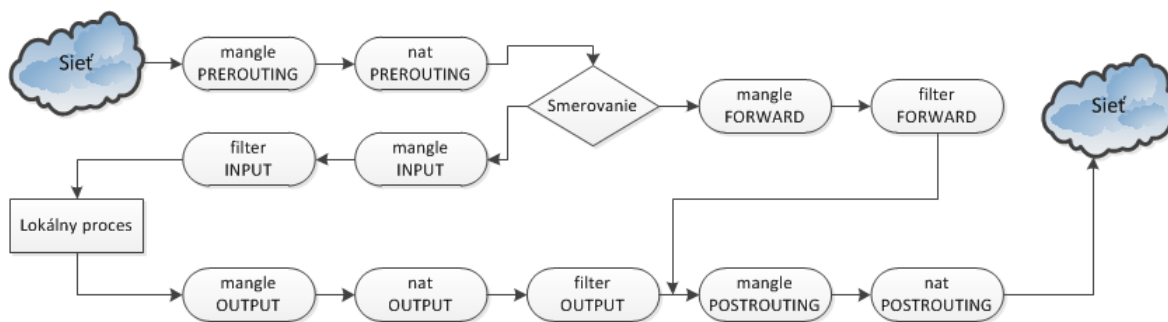
Cesta, ktorú vykonáva packet prichádzajúci do firewallu, ktorý tvorí program Netfilter nie je triviálna. Môžu nastať dva prípady – packet prichádza zo siete, alebo pochádza z lokálneho systému.

Ak packet prichádza zo siete najskôr prechádza tabuľkou **mangle** a jej reťazou *PREROUTING*. Hneď potom tabuľkou **nat** a jej reťazou *PREROUTING*. Ďalej ako vidíme aj na obrázku (Obr. 3) dochádza k smerovaniu a môžu nastať dve možnosti:

- packet je určený pre firewall, prejde tabuľkami **mangle** a **filter** a ich reťazami *INPUT* a končí v lokálnom procese,
- packet je určený inej sieti, cez firewall len prechádza, smeruje teda na obrázku vpravo hore do tabuliek **mangle** (reťaze *FORWARD*, *POSTROUTING*), **filter** (reťaz *FORWARD*) a do tabuľky **nat** (a jej reťazou *POSTROUTING*).

Ak packet vzniká na lokálnom systéme, prechádza tabuľkami **mangle** (*OUTPUT*, *POSTROUTING*), **nat** (*OUTPUT*, *POSTROUTING*) a tabuľkou **filter** (a jej reťazou *OUTPUT*) na obrázku vľavo dole [19].

V každej z reťazí je nastavená spomínaná štandardná politika (ACCEPT, DROP), ktorú je možné zmeniť.



Obr. 3 – Cesta packetu v programe Netfilter.

3.1.2 Príkaz iptables

Iptables je utilita, pomocou ktorej sa pridávajú a upravujú pravidlá v programe Netfilter. Štruktúra príkazu je:

- *iptables --parameter (názov tabuľky názov reťaze) podmienka akcia.*

Na vykonávanie príkazov *iptables* sú potrebné práva užívateľa *root*. Príkaz začína kľúčovým slovom *iptables*, za ktorým nasleduje konkrétny príkaz spolu s ďalšími parametrami. Tabuľku špecifikujeme pomocou parametra *-t*. Veľmi dôležitou súčasťou príkazu je znak výkričník, ktorým dosiahneme negáciu zadanej hodnoty. Pomocou podmienky určujeme, ktorých packetov sa daný príkaz týka. Akcia určuje, čo sa s packetom stane, ak vyhovuje podmienke – najčastejšie ACCEPT, DROP. Je možno však použiť REJECT, DENY alebo LOG.

4 E-MAIL

E-mail je jedna z celosvetovo najvyužívanějších služieb dnešného internetu, používaná dennodenne na celom svete a je dominantou komunikácie v profesionálnom styku. E-mail ako taký nebol vynájdený, ale sa postupne vyvíjal z veľmi jednoduchých začiatkov.

Prvým systémom, ktorý dokázal doručovať správy v rámci jedného počítača bol program MAILBOX používaný od roku 1965 na MIT (Massachusetts Institute of Technology). Ďalším takýmto programom bol SNDMSG (Send Message). S rozvojom internetu sa doručovanie e-mailov stávalo čoraz väčším problémom. Ray Tomlinson sa považuje za človeka, ktorý dal v roku 1972 e-mailu podobu ako ju poznáme dnes. Vybral symbol @ a navrhol pomocou neho identifikovať posielanie správy od jedného počítača na druhý. Od roku 1974 začali e-mail používať stovky vojenských užívateľov, čo bolo záchranou pre americkú vojenskú sieť ARPANET (Advanced Research Projects Agency Network). Obľuba e-mailu bola čoraz väčšia aj vďaka Larrymu Robertsovi, ktorý pre svojho nadriadeného vymyslel systém priečinkov, vďaka ktorým si mohol organizovať svoje e-maily. John Vital v roku 1975 vyvinul prvý softvér, ktorý slúžil na organizáciu e-mailov a o niekoľko rokov neskôr e-maily tvorili skoro 75% prenášaných dát v sieti ARPANET.

4.1 Ako funguje e-mail

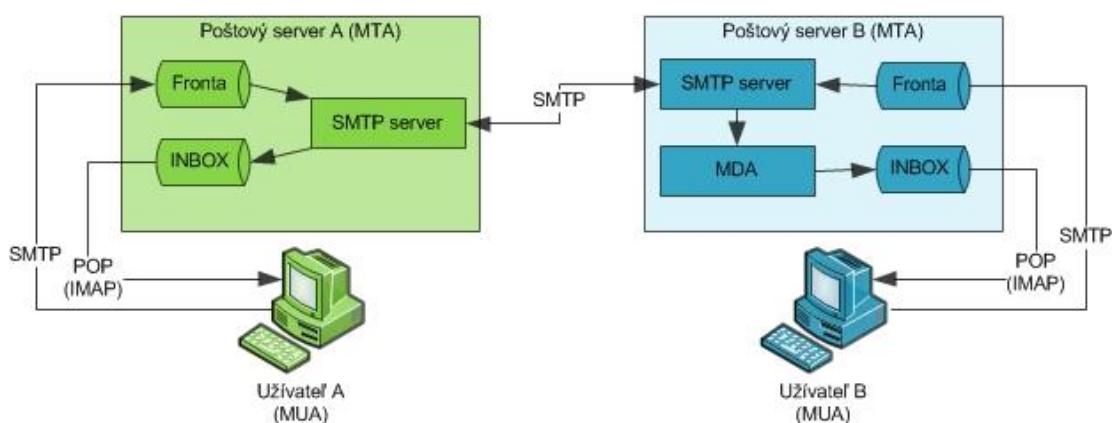
Keď uvažujeme o fungovaní elektronickej pošty, je veľmi dôležité vysvetliť tieto pojmy:

- MUA (Mail User Agent) – klientský software určený na vytváranie, spracovanie a posielanie elektronických správ.
- MTA (Mail Transfer Agent) – server, ktorý prijíma a doručuje elektronicкую poštu.
- MDA (Mail Delivery Agent) – stará sa o konečné doručenie elektronickej pošty lokálnym užívateľom.

Keď napíšeme a pošleme e-mail prostredníctvom MUA, tento program odovzdá našu správu serveru MTA. Tento sa stará o presun správ medzi jednotlivými MTA na ceste našej správy k cieľu. V prípade že MTA správu doručiť nevie, vráti správu späť odosielateľovi. Správa, ktorá dorazí na cieľový MTA je priamo doručená alebo je odovzdaná agentovi MDA, ktorý zaistí jej finálne doručenie konkrétnemu užívateľovi. Agent MDA vykonáva rôzne kontroly správy a rôzne iné špeciálne činnosti. Po uložení do

úložiska, správa zostáva v úložisku, pokiaľ si ju konečný príjemca nevyzdvihne prostredníctvom svojho MUA. Cestu e-mailu od odosielateľa k príjemcovi znázorňuje obrázok (Obr. 4).

Na odosielanie pošty dnes využívame protokol SMTP (Simple Mail Transfer Protocol), ale pre príjem pošty tento protokol nie je vhodný a preto boli vyvinuté dva protokoly POP (Post Office Protocol) a IMAP (Internet Message Access Protocol) v súčasnosti používané vo verziách POP3 a IMAP4.



Obr. 4 – Moderný e-mail.

4.2 E-mailová správa a e-mailová adresa

Očíslované dokumenty vysvetľujúce nejakú technológiu alebo protokol spravuje organizácia IETF (Internet Engineering Task Force) a vydáva ich ako RFC (Request For Comments). E-mailová správa je popísaná štandardom RFC-2822, ktorý je doplnkom pôvodného štandardu RFC-822 z roku 1982. Podľa tohto dokumentu je e-mailovou správou postupnosť znakov v rozmedzí 1 až 127 zo znakovkej sady ASCII (American Standard Code for Information Interchange) a skladá sa zo záhlavia a tela správy. Záhlavie tvoria hlavičky oddelené koncom riadku CRLF (Carriage Return Line Feed). Podobne ako v prípade klasickej pošty sú v nej uvedené informácie o odosielateľovi (From:), informácie o príjemcovi (To:), dátum a čas odoslania správy (Date:) a predmet (Subject:). Ďalšie doplnkové informácie v hlavičke sú uvedené v časti prijaté (Received:), ktorá obsahuje celú cestu kadiaľ správa putovala. Telo správy je postupnosť znakov, ktoré nasledujú za záhlavím, od ktorého je oddelené prázdny riadkom – CRLF [8].

Štandard RFC-2822 popisuje takisto veľmi podrobne aký formát by mala mať e-mailová adresa. Zjednodušene môžeme povedať, že e-mailová adresa sa skladá z troch častí:

- miestna časť –obvykle meno užívateľa, prípadne názov používateľského účtu,
- oddeľovač – symbol @,
- doména – doména, do ktorej užívateľ patrí.

Doménová časť adresy identifikuje bod – v tomto prípade e-mailový server, kde má byť naša správa doručená. Miestna časť sa často nazýva aj LHS (Left Hand Side) e-mailovej adresy a doména ako RHS (Right Hand Side).

4.3 E-mailové protokoly

4.3.1 SMTP a ESMTP

Protokol SMTP je veľmi jednoduchý protokol, ktorý sa skladá z jednoduchých štvorznakových príkazov klienta, na ktoré server odpovedá odpoveďou, ktorá obsahuje trojmiestny stavový kód nasledovaný popisom chyby. Stavové kódy servera môžeme rozdeliť na štyri triedy, ktoré vidíme v tabuľke (Tab. 6). Protokol SMTP štandardne pracuje nad protokolom TCP (Transmission Control Protocol) a používa port z triedy vyhradených „well-known ports“ s číslom 25. Rozšírenou verziou SMTP je ESMTP (Extended SMTP), ktorý dopĺňa funkcionality pôvodného protokolu. Pri SMTP sa klient serveru predstavuje pomocou príkazu *helo*, pri ESMTP je to príkaz *ehlo*, pričom môžu nastať dve situácie:

- server je len SMTP, a príkaz *ehlo* nepozná, vtedy použijeme *helo*,
- server odpovie kódom 250 a vypíše, ktoré rozširujúce príkazy podporuje.

Najznámejšie a najrozšírenejšie rozšírenia protokolu SMTP sú:

- **VERB (Verbose)** – server začne vypisovať podrobný protokol o komunikácii.
- **BITMIME** – rozšírenie, ktoré je určené na prenos správ MIME (Multipurpose Internet Mail Extension). Umožňuje posielat' spolu s e-mailom široké spektrum príloh.
- **SIZE** – špecifikuje dĺžku správy. Server ho vypisuje spolu s číselným údajom, ktorý nám hovorí akú maximálnu veľkosť správy server akceptuje.
- **ETRN** – pre malé firmy, ktoré majú poštový server za komutovanou linkou.

- **DSN (Delivery Status Notification)** – notifikácia o doručení správy.

Tab. 6 – Triedy stavových kódov protokolu SMTP.

Úroveň kódu	Popis
2xx	Príkaz bol úspešne vykonaný.
3xx	Príkaz bol prijatý, ale server očakáva od klienta ďalšie informácie.
4xx	Príkaz nebol úspešne vykonaný z dôvodu dočasného problému.
5xx	Príkaz nebol úspešný a problém je považovaný za trvalý.

4.3.2 POP3

Klient nadväzuje spojenie so serverom POP prostredníctvom TCP spojenia na porte 110. Po úspešnom nadviazaní spojenia sa server predstaví a čaká na autentizáciu užívateľa. „Základná je autentizácia menom a heslom užívateľa. V prípade, že autentizácia prebehne s kladným výsledkom, komunikácia prechádza do transakčného stavu, kedy klient môže so správami vo svojej poštovej schránke na serveri. I keď klient napr. zruší niektoré správy vo svojej poštovej schránke na serveri, behom transakčného stavu je takéto zrušenie ešte odvolateľné. Na záver relácie prejde klient do stavu UPDATE, kedy sa prevedú všetky zmeny v jeho poštovej schránke na serveri trvale [9].“

Protokol POP3 umožňuje, aby bol k poštovej schránke prihlásený používateľ iba raz. Použitie transakčnej práce so schránkou je možné vďaka jednoduchému triku. Po prihlásení klienta sa celá jeho schránka skopíruje, klient pracuje so skopírovanou schránkou a do pôvodnej stále môžu prichádzať e-maily. Končným spojením oboch schránok je spomínaný príkaz UPDATE.

4.3.3 IMAP4

Protokol IMAP je oveľa zložitejší ako dva predchádzajúce protokoly. Je určený k práci s poštovou schránkou na serveri v režimoch online i offline z viacerých aplikácií. Spojenie so serverom je nadväzované prostredníctvom TCP na porte 143. Behom práce jednej aplikácie s poštovou schránkou môže iná aplikácia zmeniť obsah stránky, môže do schránky e-mail zapísať, prípadne vymazať.

„Príkazy sa opäť zadávajú v ASCII (obdobne ako v prípade protokolov SMTP, POP3 apod.). Príkazy sú však odlišné od príkazov protokolu POP3. Odlišnosť nie je len v názvoch príkazov, ale najmä v ich filozofii. V protokole IMAP4 môže byť zadávaná rada príkazov a odpovede na ne môžu prichádzať zo serveru v ľubovoľnom poradí. Preto klient príkazy čísluje a server do svojej odpovede zopakuje číslo príkazu, na ktorý odpovedá. Je vecou klienta, ako príkazy identifikuje (čísluje) [9].“

IMAP je vhodný najmä pre ľudí, ktorí často striedajú počítače nakoľko všetky správy a zmeny v e-mailovom priečinku ostávajú uložené na serveri.

4.4 Smerovanie e-mailov

Systém DNS (Domain Name System) je veľmi dôležitým systémom, bez ktorého by využívanie internetu nebolo také komfortné a nemenej dôležitú úlohu hrá aj pri doručovaní elektronickej pošty.

Každá doména by mala mať aspoň dva autoritatívne menné servery, ktoré majú prístup k databáze s informáciami o doméne. Doména obsahuje rôzne druhy záznamov, ktoré nazývame záznamy prostriedkov, pričom z pohľadu e-mailového servera sú najdôležitejšie:

- Záznam typu A – mapuje názov na adresu protokolu IP. Obsahuje názov hostiteľa v danej doméne a jeho IP adresu.
- Záznam typu MX – sú používané pre smerovanie e-mailov. Hovoria o tom, ktoré servery v danej doméne plnia funkciu tzv. mail exchangers – servery, ktoré sa starajú o e-maily pre danú doménu. Serverov typu MX môže byť viac, jeden slúži ako primárny, ostatné plnia úlohu sekundárnych – záložných serverov.

Pri konfigurácii záznamov MX pre danú doménu je potrebná veľká obozretnosť, existuje však niekoľko doporučení, ktoré by sme mali dodržať :

- poštové servery musia mať platné záznamy typu A,
- poštové servery nesmú byť záznamy typu CNAME,
- pre poštové servery sa doporučuje používať názvy hostiteľov, nie adresy IP,
- je potrebné zadať hodnoty preferencie jednotlivých poštových serverov [2].

5 DORUČOVANIE A PRIJÍMANIE POŠTY

5.1 Doručovanie pošty - server Postfix

„Postfix napísal Wietse Venema, ktorý je známy najmä vďaka svojim bezpečnostným nástrojom a dokumentom popisujúcim zabezpečenie. Program bol sprístupnený ako software s otvoreným zdrojovým kódom v decembri 1998. Pôvodné uvedenie sponzorovala firma IBM Research, ktorá podporuje i jeho neustály vývoj [2].“

Postfix je navrhnutý modulárne, o rôzne úlohy sa v ňom starajú rôzne špecializované programy. Má viacero výhod, vďaka ktorým je považovaný za veľmi spoľahlivý a ľahko konfigurovateľný MTA a sú to hlavne jeho zabezpečenie, výkon a flexibilita.

Postfix vystupuje v e-mailovej komunikácii ako agent MTA. Stará sa o vymieňanie správ s ďalšími agentmi MTA alebo doručuje správy v rámci lokálneho systému.

5.1.1 Zložky Postfixu

Postfix tvorí viacero častí a modulov. Jeho architektúra nie je monolitická ako tomu bolo napr. v prípade programu Sendmail, ktorý využíval pri spracovaní elektronickej pošty jeden rozsiahly program. Postfix rozdeľuje úlohy medzi rôzne programy starajúce sa o konkrétne činnosti. Tieto programy sú väčšinou démony bežiacie na pozadí systému. Najdôležitejší z nich je démon *master* – riadiaci démon. Tento spúšťa ostatné démony, ktoré sa po úspešnom vykonaní úlohy ukončia. Medzi najdôležitejšie postfixové démony by sme mohli zaradiť:

- **Master** – je riadiaci démon Postfixu. Je trvale rezidentný, riadi všetky ostatné postfixové démony a podľa typu prichádzajúcej úlohy spúšťa podriadené démony.
- **Trivial-rewrite** – poskytuje svoje služby démonovi *cleanup* a prepisuje neštandardné cieľové adresy do štandardného tvaru užívateľ@fqdn.
- **Qmgr** – spravuje e-mailové fronty. Je srdcom postfixového e-mailového systému. Rozdeľuje úlohy na doručovanie e-mailov pre démonov *local*, *smtp*, *lmtp* a *pipe*.
- **Smtplib** – démon *smtplib* prenáša správy do vzdialených cieľov.
- **Smtplibd** – spracováva komunikáciu s poštovými klientmi v sieti, ktorí doručujú Postfixu správy prostredníctvom protokolu SMTP.

- **Cleanup** – spracováva novú správu. Dopĺňa chýbajúce hlavičky, prepisuje cieľové adresy pomocou démona *trivial-rewrite*. Následne umiestni správu do fronty prichádzajúcich správ.

Fronty správ sú obvykle sústredené v adresári */var/spool/postfix/*. Tento adresár môžeme meniť prostredníctvom konfiguračného parametra *queue_directory* v konfiguračnom súbore */etc/postfix/main.cf*. V tomto adresári má každá fronta svoj podadresár s menom fronty:

- **incoming** – prichádzajúca, sú do nej umiestňované všetky prichádzajúce nové správy,
- **maildrop** – odchádzajúca, obsahuje správy odoslané príkazom *sendmail*, ktoré neboli do Postfixu poslané programom *pickup*,
- **deferred** – odložená, obsahuje správy, ktorých príjemcovia sú z nejakého dôvodu nedostupní,
- **active** – aktívna, správy, ktoré sú pripravené na odoslanie, avšak ešte odoslané neboli,
- **hold** – zadržaná, správy z nejakého dôvodu definovaného správcom vyradené z normálneho spracovania,
- **corrupt** – poškodená, obsahuje poškodené súbory.

5.1.2 Obecná konfigurácia

Po inštalácii servera Postfix a pri jeho konfigurácii budeme najviac používať adresáre:

- */etc/postfix* - umiestnenie konfiguračných súborov,
- */var/spool/postfix* - súbory front popisovaných vyššie.

Najvyužívanéjšie súbory pri konfigurácii sú:

- */etc/postfix/main.cf* - hlavný konfiguračný súbor servera,
- */etc/postfix/master.cf* - konfiguračný súbor démonov popisovaných vyššie.

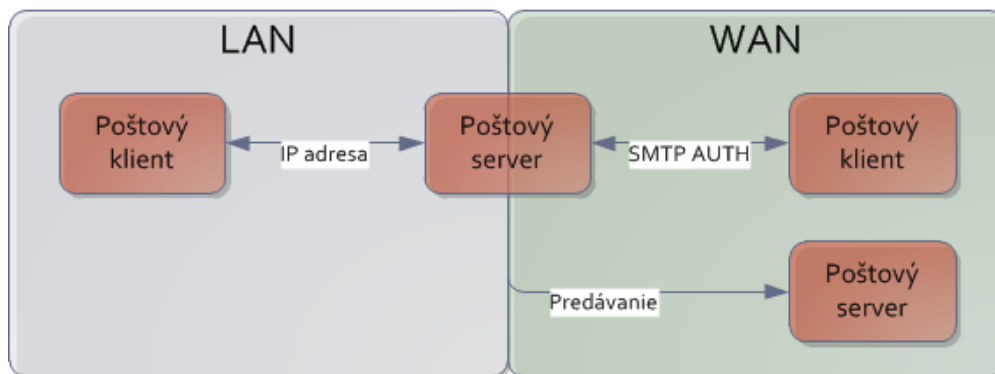
Súbor *main.cf* je možné editovať aj príkazom *postconf* alebo predvoleným textovým editorom. Všetky príkazy, ktoré majú vplyv na činnosť servera Postfix nesmú začínať

znakom komentára „#“. Parametre môžu byť zapísane v ľubovoľnom poradí a zapisujú sa spôsobom: *parameter* = *hodnota*. Jedinou podmienkou je, že definícia parametru musí začínať v prvom stĺpci konfiguračného súboru. Na hodnotu skôr definovaného parametra sa môžeme odkazovať pomocou znaku „\$“. Samozrejmosťou je možnosť použiť ako hodnotu parametra viac hodnôt, vtedy môžeme jednotlivé hodnoty oddeľovať čiarkou, medzerami, tabulátormi alebo novými riadkami. Po každej zmene konfiguračného súboru *main.cf* musíme znovu načítať všetky konfiguračné hodnoty pomocou príkazu *postfix reload*.

5.1.3 Autentifikácia SMTP

Protokol SMTP je v základe veľmi jednoduchý protokol. Autentifikácia SMTP je spôsob ako identifikovať poštových klientov. Vďaka tomuto mechanizmu umožňujeme posilať poštu aj takým klientom, ktorých IP adresy sú pre náš poštový server nedôveryhodné (Obr. 5). Táto funkcionálna sa zavádza po rozmachu spamu v internetovej pošte. Klientov, ktorí chcú prostredníctvom nášho serveru posilať spam vďaka tomuto mechanizmu odmietame. Ďalším problémom, ktorý môžeme vyriešiť vďaka autentifikácii SMTP sú mobilní klienti, ktorí potrebujú mať prístup k prostriedkom poštového servera kdekoľvek z internetu. V našom prípade využijeme metódu nazvanú SMTP AUTH, ktorú môžeme popísať nasledovne:

1. Server SMTP ponúkne SMTP AUTH klientovi.
2. Klient pošle svoje autentizačné údaje.
3. Na základe týchto údajov je povolené alebo zakázané posielanie pošty.



Obr. 5 – Princíp autentifikácie SMTP.

Pre identifikáciu užívateľov, ktorí chcú používať náš poštový server je v Postfixe implementovaný mechanizmus SASL (Simple Authentication and Security Layer). Je to metóda ako do nezabezpečeného protokolu SMTP pridať podporu autentifikácie a umožniť tak klientom overiť sa voči serveru pomocou protokolu SASL. Server Postfix na doplnenie tejto funkcionality využíva knižnice systému Cyrus-SASL. SASL sa skladá z troch vrstiev, ktoré musíme nakonfigurovať:

- autentizačné rozhranie,
- overovací mechanizmus – ANONYMOUS, CRAM-MD5, DIGEST-MD5, PLAIN, LOGIN, OTP,
- overovacia metóda – saslauthd, auxprop.

5.2 Prijímanie pošty – server Dovecot

V časti 4.3 sme hovorili o protokoloch POP3 a IMAP4, ktoré umožňujú užívateľom pracovať s e-mailovými správami uloženými na e-mailovom serveri. Server Postfix neimplementuje tieto dva protokoly, len správy doručuje a tie sú potom predávané serveru POP/IMAP, ktorých existuje veľké množstvo. Správy sú uložené do úložiska správ, pričom server Postfix rozoznáva formát Mbox a Maildir (implicitne sa používa Mbox).

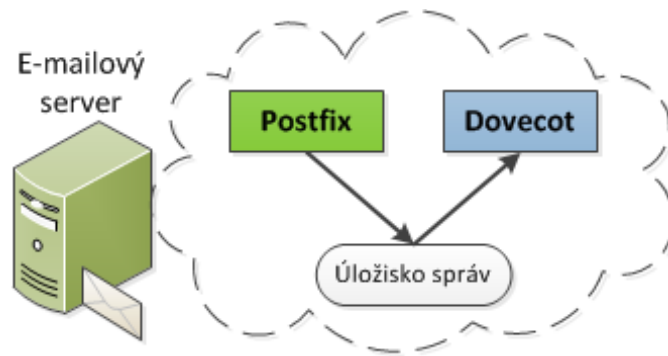
Oba tieto formáty majú rôzne výhody aj nevýhody. Pri hľadaní a sprístupňovaní je pravdepodobne rýchlejší Maildir, zatiaľ čo pri doručovaní správ je rýchlejší formát Mbox.

5.2.1 POP/IMAP server Dovecot

Po doručení správ je potrebné, aby užívatelia mali k svojim správam prístup aj z počítačovej siete. Na túto činnosť sa využívajú servery POP/IMAP.

Veľmi obľúbeným a pomerne mladým serverom určeným na túto činnosť je server Dovecot. Podporuje formát Mbox a Maildir a pri jeho tvorbe kladli autori veľký dôraz na bezpečnosť a funkcionality. Dovecot umožňuje robiť väčšie množstvo vecí ako väčšina používaných POP/IMAP serverov [13].

Princíp činnosti POP/IMAP servera je veľmi jednoduchý a môžeme ho vidieť na obrázku (Obr. 6). Keď server Postfix doručí správu, uloží ju do lokálneho úložiska správ. POP/IMAP server správu prečíta priamo z tohto úložiska a zobrazí ju užívateľovi.



Obr. 6 – Server POP/IMAP.

5.2.2 Obecná konfigurácia

Po inštalácii servera Dovecot je cesta k jeho hlavnému konfiguračnému súboru väčšinou nasledovná: `/etc/dovecot.conf`. V prípade, že daný súbor nenájdeme v tomto umiestnení, môžeme použiť príkaz:

```
[root@pluto /]# dovecot -n | head -1  
# 2.0.9: /etc/dovecot/dovecot.conf
```

Tak ako v prípade Postfixu, všetky riadky v konfiguračnom súbore začínajúce znakom „#“ sú považované za komentár. Medzery a znaky tabulátora sú ignorované. Definícia parametrov sa ako v prípade Postfixu zapisujú štýlom *parameter = hodnota*. Parameter môže obsahovať viac hodnôt, pričom sú oddelené medzerou.

II. PRAKTICKÁ ČASŤ

6 ANALÝZA VYTVÁRANÉHO RIEŠENIA

Dôvodom pre vytvorenie nového e-mailového servera bolo zastarané riešenie e-mailovej komunikácie, ktoré už nepostačovalo požiadavkám. Server už nespĺňa požiadavky moderného e-mailového servera, preto bol vytvorený e-mailový server, ktorý dokáže obslúžiť poštovú komunikáciu malej firmy, pričom sa nevylučuje ani použitie vo väčšom prostredí.

SMTP server by mal podporovať autentifikáciu klientov a umožňovať užívateľom spravovať e-maily prostredníctvom klienta MUA ako aj pomocou webmailu. E-mailová komunikácia by mala byť zabezpečená, kontrolovaná na spam. Ďalšou požiadavkou na server je, aby umožnil pripojenie malej siete do Internetu pomocou prekladu lokálnych IP adries na verejne smerovateľné IP adresy pomocou NAT. Na serveri by mal byť sprevádzkovaný firewall pomocou programu Netfilter. Ďalšou požiadavkou je konfigurácia grafického nástroja na správu servera.

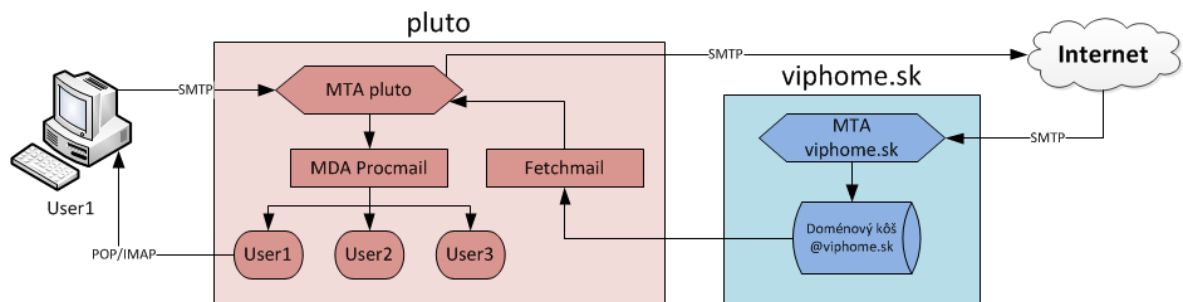
6.1 Návrh riešenia

Po analýze požiadaviek boli pri riešení použité nasledovné nástroje a programy:

- SMTP server – *Postfix*.
- POP/IMAP server – *Dovecot*.
- Webmail – *SquirrelMail*.
- Kontrola spamu a antivírus – *ClamAV*, *MailScanner*, *SpamAssassin*.
- Firewall – *Netfilter*.
- Grafická správa serveru – *Webmin*.

U prevádzkovateľa slovenských domén bola zaregistrovaná doména s adresou *www.viphome.sk*. Na prevádzku e-mailového spojenia bol zriadený u tohto poskytovateľa doménový kôš – jedna poštová schránka, ktorá bude prijímať e-maily pre celú doménu. Dôležitým parametrom teda nie je meno príjemcu, ale doména príjemcu. Môžeme povedať, že doménový kôš je jeden účet pre neobmedzený počet užívateľov. Správy z doménového koša chceme ukladať a pracovať s nimi na našom serveri, využitý bol preto program

Fetchmail, ktorý správy presunie na náš poštový server pomocou protokolu POP3. Navrhnuté riešenie vidíme na obrázku (Obr. 7).



Obr. 7 – Riešenie e-mailového servera.

Umiestnenie firewallu bolo navrhnuté na rovnakom serveri ako všetky ostatné popisované služby. Podľa požiadaviek boli povolené tieto služby a rozdelené na povolené služby pre sieť LAN (Local Area Network) a povolené služby pre sieť WAN (Wide Area Network):

- povolené služby LAN: *http, https, smtp, pop3, imap, ftp, icq, skype, dns, ssh, icmp, 10000,*
- povolené služby WAN: *ssh, http, 10000.*

Číslo 10 000 je port, na ktorom počúva webové grafické rozhranie programu Webmin určeného na grafickú správu servera.

7 INŠTALÁCIA A KONFIGURÁCIA SYSTÉMU

Inštalovaným systémom je Red Hat Enterprise Linux. Pri inštalácii boli štandardne vytvorení dvaja užívatelia:

- *root* – práva super užívateľa,
- *michal* – štandardný užívateľ systému.

Server bol pomenovaný ako niekdajšia posledná planéta našej slnečnej sústavy – Pluto. Ďalšie kroky, ktoré boli vykonané pri inštalácii, konfigurácia siete a iných neštandardných parametrov servera sú popísané ďalej.

7.1 Konfigurácia RAID, LVM a rozdelenie diskov

Treba uviesť na pravú mieru, že v tomto prípade myslíme konfiguráciu softwarového RAID. Existuje aj hardwarový RAID, na ktorý je potrebný špecializovaný hardware, ktorý môže byť v podobe samostatnej RAID karty alebo v súčasnosti častejším riešením integrovaný čip na základnej doske.

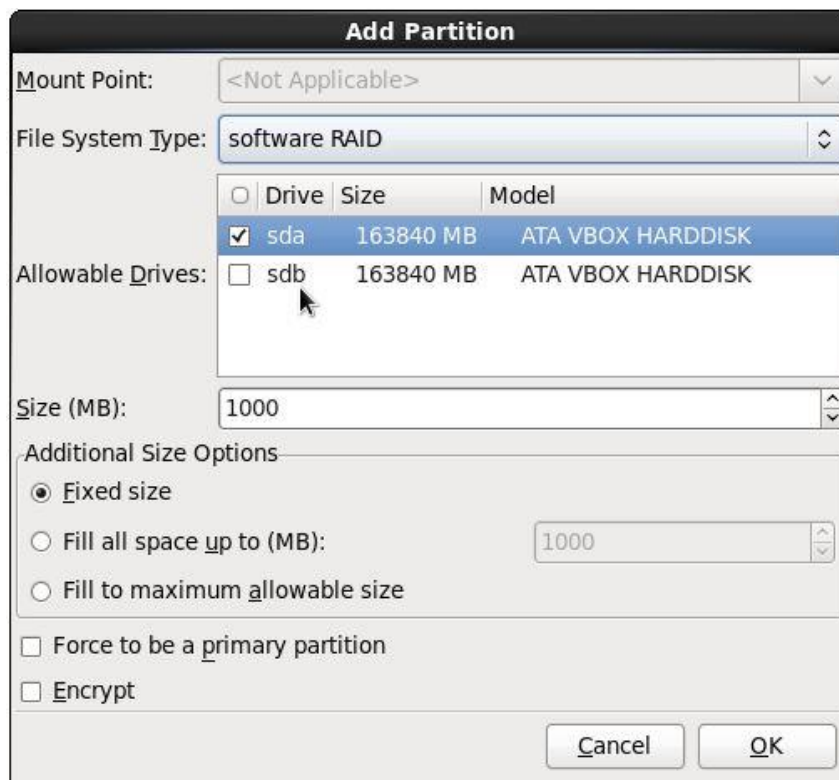
Server obsahuje dva 160 GB disky. Samotný RAID bol vytvorený počas inštalácie systému. Pomocou inštalačného sprievodcu bolo vytvorené RAID 1 pole a nad týmto RAID poľom oddiel LVM. Postup je popísaný nižšie.

Pri inštalácii postupujeme podľa krokov inštalačného sprievodcu a pri výbere akým spôsobom sa má systém Linux nainštalovať na lokálne disky vyberieme „**Use Custom Layout**“. Na oboch diskoch vytvoríme najskôr RAID partície – vpravo dole klikneme „**Create**“ a v ponúknutom dialógovom okne označíme „**RAID Partition**“. Podľa obrázku (Obr. 8) vytvoríme RAID partíciu o veľkosti 1 GB najskôr na prvom fyzickom disku a potom na druhom fyzickom disku podľa predchádzajúceho postupu. Po dokončení tohto kroku budeme mať dve RAID partície, z ktorých môžeme vytvoriť RAID zariadenie. Opäť klikneme na voľbu „**Create**“ a vyberieme „**RAID Device**“. V ponúknutom dialógovom okne vyberieme postupne od vrchu nadol:

- **Mount Point:** /boot,
- **File System Type:** ext4,
- **RAID Device:** md0,

- **RAID Level:** RAID 1.

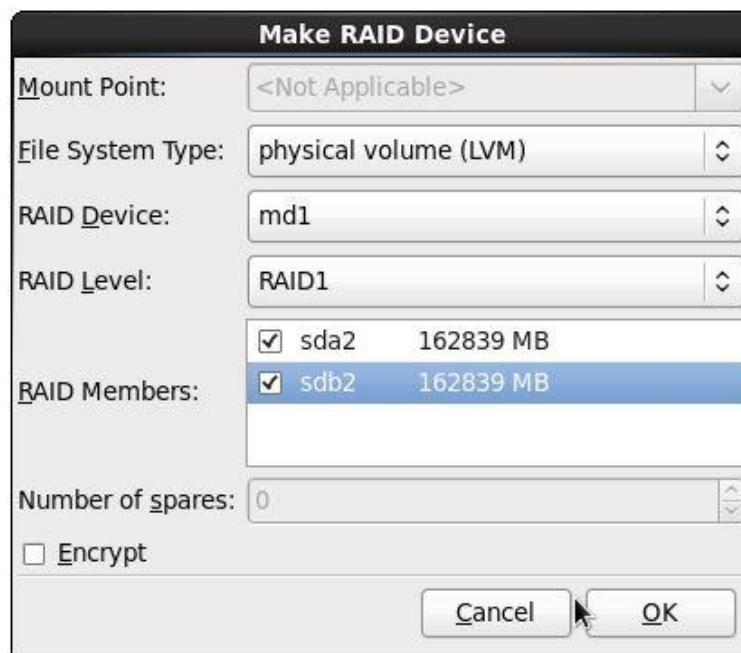
Označíme obe RAID partície, ktoré boli vytvorené podľa postupu vyššie a klikneme na „OK“. Týmto bolo vytvorené RAID 1 zariadenie, na ktorom bude uložený súborový systém */boot*.



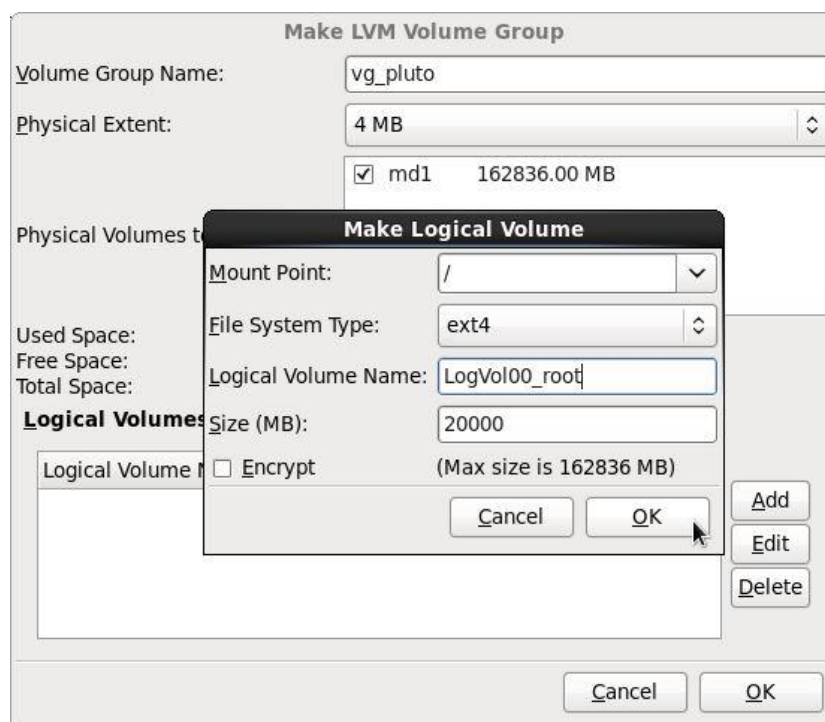
Obr. 8 – Vytvorenie RAID partície.

Ďalším krokom bolo vytvorenie LVM nad RAID partíciou. Podľa postupu popísaného vyššie boli vytvorené dve RAID partície tak, že zapĺňali celé zvyšné miesto na oboch diskoch. Nad vzniknutými partíciami bol vytvorený oddiel LVM.

Vpravo dole vyberieme „Create“ – „RAID Device“ voľby však nastavíme podľa obrázku (Obr. 9). Týmto bola vytvorená partícia nad RAID 1, ktorá je označená ako „Physical volume“. Nad týmto vytvoreným zväzkom bola vytvorená LVM Volume group - „Create“ – „LVM Volume Group“, ktorá bola rozdelená na jednotlivé logické zväzky podľa rozdelenia z časti 2.1.3. Vytváranie jednotlivých súborových systémov vidíme na obrázku (Obr. 10).



Obr. 9 – Vytvorenie LVM fyzického zväzku.



Obr. 10 – Vytváranie jednotlivých logických zväzkov LVM.

Po tomto kroku inštalácie by rozdelenie diskov malo zodpovedať obrázku (Obr. 11). Ďalej pokračujeme v inštalácii štandardným spôsobom a nainštalujeme systém.

Device	Size (MB)	Mount Point/ RAID/Volume	Type	Format
LVM Volume Groups				
vg_pluto	162836			
LogVol00_root	20000	/	ext4	✓
LogVol01_home	20000	/home	ext4	✓
LogVol04_mail	30000	/var/spool/mail	ext4	✓
LogVol03_var_log	10000	/var/log	ext4	✓
LogVol06_swap	2048		swap	✓
LogVol05_tmp	10000	/tmp	ext4	✓
LogVol02_var	10000	/var	ext4	✓
Free	60788			
RAID Devices				
md0 (/dev/md0)	998	/boot	ext4	✓
md1 (/dev/md1)	162837	vg_pluto	physical volume (LVM)	✓
Hard Drives				
sda (/dev/sda)				
sda1	1000	md0	software RAID	✓
sda2	162839	md1	software RAID	✓
sdb (/dev/sdb)				
sdb1	1000	md0	software RAID	✓
sdb2	162839	md1	software RAID	✓

Obr. 11 – Konfigurácia RAID, LVM, rozdelenie partícií.

Toto rozdelenie a použitie LVM má v prípade použitia na serveri svoje opodstatnenie. V prípade, že nám na nejakom logickom oddieli v rámci LVM Volume group dochádza miesto, jednoducho ho rozšírime o ďalšie voľné miesto z tejto volume group. Nemusíme server vypínať, reštartovať ani čokoľvek iné, urobíme to jednoducho za behu operačného systému.

Informácie o physical volume, ktorý bol vytvorený pri inštalácii nad RAID 1 partíciou môžeme zistiť príkazom:

```
[root@pluto /]# pvddisplay
```

Ďalej si môžeme vypísať informácie o skupine logických zväzkov (volume group), ktoré boli taktiež vytvorené pri inštalácii a to pomocou príkazu:

```
[root@pluto /]# vgdisplay
```

Skrátenie výpisu.

```
Format                lvm2
```

Skrátenie výpisu

VG Access *read/write*

VG Status *resizable*

Skrátenie výpisu

VG Size *159.02 GiB*

PE Size *4.00 MiB*

Total PE *40709*

Alloc PE / Size *25512 / 99.66 GiB*

Free PE / Size *15197 / 59.36 GiB*

Z výpisu je zrejme použité formátovanie – lvm2. Veľmi dôležitou časťou vo výpise je skratka „**PE / Size**“ (physical extent), ktorá nám napovedá ako LVM funguje. Rozdeľuje fyzické zväzky do blokov určitej veľkosti PE, ktoré potom prideliť jednotlivým logickým zväzkom. V tomto prípade má jeden PE veľkosť 4 MB a môžeme vidieť, že ich máme k dispozícii 40 709 pričom 25 2512 (99.66 GB) bolo použitých a ďalších 15 197 (59.36 GB) bolo k dispozícii pre ďalšie logické zväzky.

7.1.1 Vytvorenie logického zväzku

V prípade, že na skupine zväzkov máme k dispozícii ešte nejaké miesto, môžeme pristúpiť k vytvoreniu ďalšieho logického zväzku. Použijeme príkaz:

```
[root@pluto /]# lvcreate -L 7G vg_pluto -n ZALOHA
```

V tomto prípade pomocou príkazu *lvcreate -L* vložíme požadovanú veľkosť logického zväzku, názov volume group, na ktorom chceme vytvoriť logický zväzok (*vg_pluto*) a jeho názov pomocou parametra *-n* (ZALOHA).

Nový logický zväzok môžeme pripojiť do pripraveného adresára alebo niekde do súborového systému, najskôr ho však musíme naformátovať pomocou príkazu:

```
[root@pluto /]# mkfs.ext4 /dev/vg_pluto/ZALOHA
```

Pripojíme ho do predpripraveného adresára */mnt* pomocou príkazu:

```
[root@pluto /]# mount dev/vg_pluto/ZALOHA /mnt
```

7.1.2 Zväčšenie logického zväzku

Ak nám na nejakom logickom zväzku v rámci volume group dochádza miesto, je veľmi jednoduché tento zväzok rozšíriť o ďalšie miesto použitím príkazu:

```
[root@pluto /]# lvextend -L +1G /dev/vg_pluto/ZALOHA
```

Keď sa však príkazom *df -h* presvedčíme, či sa logický zväzok zväčšil, vidíme, že má stále veľkosť 7GB.

```
[root@pluto /]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/vg_pluto-ZALOHA	6.9G	144M	6.4G	3%	/mnt

Logický oddiel sa chová ako diskový oddiel, ktorý bol rozšírený, ale súborový systém na ňom ostal bez zmeny, teda má rovnakú kapacitu ako pred rozšírením. Aby súborový systém vzniknuté miesto využil, treba priestor rozšíriť ručne. Na to môžeme použiť program *resize2fs* a použiť príkaz:

```
[root@pluto /]# resize2fs /dev/vg_pluto/ZALOHA
```

Po opätovnom použití príkazu *df -h* je zrejmé, že vzniknutý logický zväzok má správnu veľkosť:

```
[root@pluto /]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/vg_pluto-ZALOHA	7.9G	144M	7.4G	2%	/mnt

7.1.3 Zmenšenie logického zväzku

Redukovať veľkosť pripojeného súborového systému za behu systému nebolo možné, preto bolo nutné najskôr daný logický zväzok, ktorý chceme zmenšiť odpojiť. Nasleduje zmenšenie súborového systému pomocou príkazu *resize2fs*. Je doporučené zmenšený súborový systém preveriť nástrojom *e2fsck*. Po kontrole a zmenšení môžeme súborový systém znovu pripojiť a zmenšiť logický zväzok príkazom *lvreduce*:

```
[root@pluto /]# umount /mnt/
```

```
[root@pluto /]# resize2fs /dev/vg_pluto/ZALOHA 6G
```

```
resize2fs 1.41.12 (17-May-2010)
```

```
Please run 'e2fsck -f /dev/vg_pluto/ZALOHA' first.
```

```
[root@pluto /]# e2fsck -f /dev/vg_pluto/ZALOHA
```

```
e2fsck 1.41.12 (17-May-2010)
```

Skrátenie výpisu

```
/dev/vg_pluto/ZALOHA: 11/524288 files (0.0% non-contiguous), 69711/2097152 blocks
```

```
[root@pluto /]# resize2fs /dev/vg_pluto/ZALOHA 6G
```

```
resize2fs 1.41.12 (17-May-2010)
```

Resizing the filesystem on /dev/vg_pluto/ZALOHA to 1572864 (4k) blocks.

The filesystem on /dev/vg_pluto/ZALOHA is now 1572864 blocks long.

```
[root@pluto /]# mount /dev/vg_pluto/ZALOHA /mnt/
```

```
[root@pluto /]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/vg_pluto-ZALOHA	6.0G	142M	5.5G	3%	/mnt

```
[root@pluto /]# lvreduce -L 6G /dev/vg_pluto/ZALOHA
```

Skrátenie výpisu

Do you really want to reduce ZALOHA? [y/n]: y

Reducing logical volume ZALOHA to 6.00 GiB

Logical volume ZALOHA successfully resized

7.2 Nastavenie logovacieho démona

Všetky logovacie súbory by sa štandardne mali nachádzať v adresári `/var/log/`. Je veľmi vhodné a doporučené umiestniť tento adresár na samostatný diskový oddiel prípadne samostatný disk tak, ako bolo popisované v časti 2.1.3, aby systémové logovanie fungovalo aj v prípade zaplnení ostatných diskov či partícií.

7.2.1 Logovanie prihlasovania

Server, ktorý bol vytvorený v tejto práci plní zároveň aj úlohu firewallu a preto sú dôležité informácie, kto a kedy sa pokúšal na server prihlasovať a podobne. Konfiguračný súbor ***rsyslog.conf***, ktorý sa nachádza v adresári `/etc/` bol upravený tak, aby sa logovali aj tieto informácie a aby boli oddelené do samostatného súboru: ***/var/log/auth.log***. V konfiguračnom súbore bol pridaný nasledovný riadok:

- `auth, authpriv.* /var/log/auth.log`

Tento riadok zabezpečí logovanie autentifikačných správ (*auth*), všetkých nesystémových autorizačných a autentifikačných správ (*authpriv.**) a všetky zapíše do súboru *auth.log*.

7.2.2 Logovanie e-mailového systému

Ďalšími dôležitými informáciami, ktoré sú dôležité, sú informácie týkajúce sa e-mailového servera. Bolo potrebné logovať všetky informácie týkajúce sa e-mailového systému a ostatné rozdelíme do viacerých súborov podľa ich dôležitosti. Do konfiguračného súboru *rsyslog.conf* boli pridané nasledovné riadky:

- *mail.** -/var/log/mail.log
- *mail.info* /var/log/mail.info
- *mail.warn* /var/log/mail.warn
- *mail.err* /var/log/mail.err

Pomlčka pred */var/log/mail.log* indikuje asynchrónny zápis do súboru a prikazuje logovaciemu démonovi, aby všetky správy týkajúce sa e-mailového systému nezapisoval na disk okamžite.

7.2.3 Reštart logovacej služby

Po zmene konfiguračného súboru *rsyslog.conf*, bolo potrebné konfiguráciu znova načítať a reštartovať démona *syslogd*. V systéme Red Hat bol tento krok vykonaný príkazmi:

```
[root@pluto /]# /etc/init.d/rsyslog reload
```

```
[root@pluto /]# service rsyslog restart
```

7.2.4 Konfigurácia rotácie logov

Ako bolo spomenuté v časti 2.6 o rotáciu logov sa stará program *logrotate* prostredníctvom svojho konfiguračného súboru *logrotate.conf*. Tento konfiguračný súbor bol ponechaný bez zmeny. Zmena bola vykonaná v čiastkových konfiguračných súboroch s názvom služby, ktorej logy treba rotovať, ktoré sú uložené v adresári */etc/logrotate.d/*. Keďže chceme rotovať logy systémovej logovacej služby *syslog* konfiguračný súbor bol upravený nasledovne:

- riadok */var/log/maillog* nahradíme riadkami:
 - */var/log/mail.log*,
 - */var/log/mail.info*,

- */var/log/mail.warn*,
- */var/log/mail.err*,
- riadok *auth.* /var/log/auth* nahradíme riadkom:
 - *auth, authpriv.* /var/log/auth.log*.

7.3 Konfigurácia siete

7.3.1 Konfigurácia mena servera

Meno servera bolo nakonfigurované manuálne zmenou parametra *HOSTNAME* na hodnotu "**pluto**" v konfiguračnom súbore */etc/sysconfig/network*. Aby sa zmena prejavila, bolo nutné systém reštartovať.

7.3.2 Konfigurácia sieťového pripojenia

Rovnako, ako v prípade mena servera môžeme postupovať aj v prípade konfigurácie sieťového pripojenia servera, ktoré môžeme nakonfigurovať už počas inštalácie systému alebo neskôr viacerými spôsobmi. Jedným z nich je zadanie príkazu *redhat-config-network* z príkazového riadku ako používateľ *root*, ktorý spustí grafického sprievodcu na konfiguráciu siete. Ďalšou možnosťou je použitie príkazu *setup*. Poslednou možnosťou je editácia konfiguračných súborov sieťových kariet, ktoré sú umiestnené */etc/sysconfig/network-scripts/*. Náš server má osadené dve sieťové karty, jednu pre vnútornú sieť (*eth0*) a jednu pre vonkajšiu sieť (*eth1*), editované boli konfiguračné skripty označené *ifcfg-eth0* a *ifcfg-eth1*. Poskytovateľ internetových služieb nám poskytol nasledovné informácie:

- verejná IP adresa – 217.75.92.249,
- sieťová maska – 255.255.255.248,
- predvolená brána – 217.75.92.254,
- DNS1 – 208.67.222.222,
- DNS2 – 208.67.220.220.

Pomocou textového editora boli oba konfiguračné súbory upravené tak, ako vidíme v tabuľke (Tab. 7).

Tab. 7 – Konfigurácia sieťových kariet servera.

ifcfg-eth0	ifcfg-eth1
DEVICE= eth0	DEVICE= eth1
BOOTPROTO= none	BOOTPROTO= none
BROADCAST= 192.168.1.255	BROADCAST= 217.75.92.255
HWADDR= 00:50:56:B2:42:AB	HWADDR= 00:50:56:C3:42:DC
IPADDR= 192.168.1.1	IPADDR= 217.75.92.249
NETMASK= 255.255.255.0	NETMASK= 255.255.255.248
ONBOOT= YES	ONBOOT= YES
TYPE= Ethernet	TYPE= Ethernet
USERCTL= NO	USERCTL= NO
IPV6INIT= NO	IPV6INIT= NO
PEERDNS= YES	PEERDNS= YES
	GATEWAY= 217.75.92.254

Súbor */etc/sysconfig/network* bol zmenený takto:

```
[root@pluto /]# cat /etc/sysconfig/network
```

```
NETWORKING=yes
```

```
HOSTNAME=pluto.viphome.sk
```

```
GATEWAY=217.75.92.254
```

Súbor */etc/resolv.conf*, ktorý poskytuje nášmu systému adresy DNS serverov od poskytovateľa bol zmenený nasledovne:

```
[root@pluto /]# cat /etc/resolv.conf
```

```
nameserver 208.67.222.222 #Primarny DNS server
```

```
nameserver 208.67.220.220 #Sekundarny DNS server
```

7.4 Vypnutie nepotrebných služieb

„Pri výstavbe nového servera sa teda musíte neustále pýtať: akú prácu budem od tohto počítača skutočne vyžadovať? Naozaj musia na webovom serveri bežať aj služby FTP? Má

na serveri DNS bežať tiež služba NFS, i keď systém neexportuj žiadne zdieľané adresáre [6]?”

Pomocou takýchto a podobných otázok si môžeme naozaj efektívne určiť, ktoré služby by mali byť na našom serveri naozaj spustené. Príkazom *chkconfig --list | grep on* môžeme vypísať všetky služby spúšťané v jednotlivých úrovniach behu nášho systému a postupne vypnúť všetky, ktoré nepotrebujeme.

Na serveri boli nepotrebné RPC (Remote Procedure Call) služby, ďalej služby Automounter a NFS. Všetky tieto boli postupne vypnuté nasledovnými príkazmi:

```
[root@pluto /]# chkconfig --level 0123456 nfslock off  
[root@pluto /]# chkconfig --level 0123456 netfs off  
[root@pluto /]# chkconfig --level 0123456 rpcgssd off  
[root@pluto /]# chkconfig --level 0123456 rpcidmapd off  
[root@pluto /]# chkconfig --level 0123456 autofs off
```

Na e-mailovom serveri je nepotrebná tlačová služba, bola preto vypnutá príkazom:

```
[root@pluto /]# chkconfig --level 0123456 cups off
```

Po vypnutí všetkých nepotrebných služieb reštartujeme server a môžeme zobrazíť počúvajúce služby spolu s portom, na ktorom server počúva prichádzajúce požiadavky príkazom:

```
[root@pluto /]# netstat -tulpn
```

8 INŠTALÁCIA A KONFIGURÁCIA POSTFIX

8.1 Inštalácia MTA Postfix

MTA Postfix môžeme do systému nainštalovať priamo pri inštalácii operačného systému. V prípade, že sme tak neurobili, môžeme Postfix nainštalovať manuálne neskôr. Pred samotnou inštaláciou bol v systéme pomocou programu *vipw* vytvorený užívateľský účet *postfix*, ktorý bude vlastníkom front Postfixu, všetkých jeho procesov, nebude umožňovať prihlásenie a nebude mať ani domovský adresár. Heslo bolo nastavené na „*“ a domovský adresár na neplatné cesty. Vyžaduje sa aj vyhradená skupina, ktorá nie je používaná žiadnym užívateľským účtom. Použitý bol program *vigr* a vytvorená skupina *postdrop*. Po stiahnutí najnovšej verzie Postfixu bola inštalácia vykonaná nasledovným spôsobom:

```
[root@pluto Downloads]# gzip -d postfix-2.0.10.tar.gz
```

```
[root@pluto Downloads]# tar -xf postfix-2.0.10.tar
```

```
[root@pluto Downloads]# cd postfix-2.0.10
```

Ak akceptujeme všetky implicitné parametre pre preklad Postfixu, môžeme pristúpiť ku kompilácii zadáním príkazu *make*. Pri inštalácii Postfixu musíme pracovať ako *root* a musíme poznať cestu k programu Sendmail a cesty k súborom *mailq* a *newaliases*. Príkazom *make install* spustíme proces inštalácie a odpovedáme na otázky inštalračného sprievodcu.

```
[root@pluto postfix-2.0.10]# make install
```

```
[root@pluto postfix-2.0.10]# install_root: [/]
```

```
[root@pluto postfix-2.0.10]# tempdir: [/home/michal/postfix-2.0.10]
```

```
[root@pluto postfix-2.0.10]# config_directory: [/etc/postfix]
```

```
[root@pluto postfix-2.0.10]# daemon_directory: [/usr/libexec/postfix]
```

```
[root@pluto postfix-2.0.10]# command_directory: [/usr/sbin]
```

```
[root@pluto postfix-2.0.10]# queue_directory: [/var/spool/postfix]
```

```
[root@pluto postfix-2.0.10]# sendmail_path: [/usr/lib/sendmail]
```

```
[root@pluto postfix-2.0.10]# newaliases_path: [/usr/bin/newaliases]
```

```
[root@pluto postfix-2.0.10]# mailq_path: [/usr/bin/mailq]
```

```
[root@pluto postfix-2.0.10]# mail_owner: [postfix]
```

```
[root@pluto postfix-2.0.10]# setgid_group: [postdrop]
```

```
[root@pluto postfix-2.0.10]# manpage_directory: [/usr/local/man]
```

```
[root@pluto postfix-2.0.10]# sample_directory: [/etc/postfix]
```

```
[root@pluto postfix-2.0.10]# readme_directory: [no]
```

Ak inštalácia prebehla, v systéme máme úspešne nainštalovaný server Postfix.

8.2 Postfix ako predvolený MTA

Predvoleným programom na prenos pošty je v systémoch Red Hat program Sendmail. Po inštalácii Postfixu bol stále za agenta na prenos pošty považovaný program Sendmail. Ako užívateľ *root* zadáme na príkazovom riadku príkaz *alternatives --config mta* a vyberieme ako predvolený program Postfix:

```
[root@pluto /]# alternatives --config mta
```

There are 2 programs that provides 'mta'.

Selection Command

```
-----
*+    1      /usr/sbin/sendmail.sendmail
```

```
      2      /usr/sbin/sendmail.postfix
```

Enter to keep the current selection[+], or type selection number:2

Po tomto kroku môžeme Postfix spúšťať pri zavádzaní systému:

```
[root@pluto /]# chkconfig --level 2345 postfix on
```

8.3 Konfigurácia Postfix

V tejto časti bol editovaný hlavný konfiguračný súbor Postfixu *main.cf*. a zmenené tieto hodnoty:

```
myhostname = pluto.viphome.sk
```

```
mydomain = viphome.sk
```

```
myorigin = $mydomain
```

```
inet_interfaces = all
```

```
inet_protocols = ipv4
```

```
mydestination = $myhostname, localhost.$mydomain, $mydomain
```

```
mynetworks = 127.0.0.0/8, 192.168.1.0/24
```

```
home_mailbox = Maildir/
```

```
mail_spool_directory = /var/spool/mail/
```

luser_relay = **mic**hal

smtpd_banner = \$myhostname **ES**SMTP \$mail_name (\$mail_version)

Pomocou týchto parametrov bola vykonaná základná konfigurácia Postfixu. V prvom parametri *myhostname* sme zadali plne kvalifikovaný názov systému, na ktorom prevádzkujeme Postfix. Parameter *mydomain* určuje doménu, ako to plynie z jeho názvu. Veľmi dôležitým parametrom je *myorigin*, ktorý určuje doménový názov pridávaný k menu užívateľského účtu pri odosielaní pošty. V našom prípade bude adresu odosielateľa tvoriť názov užívateľského účtu spolu s *mydomain*. Poštu prijímame na všetkých sieťových kartách servera prostredníctvom protokolu IPv4. Parameter *mydestination* určuje, pre ktoré domény Postfix poшту prijíma. V tomto prípade pre celú doménu. Relay – posielanie pošty bolo povolené len klientom z lokálneho systému a z tej istej podsiete v akej je samotný server – parameter *mynetworks*. Pošta bola ukladaná do domovských priečinkov užívateľov vo formáte Maildir (*home_mailbox*, *mail_spool_directory*). Toto nastavenie so sebou nesie aj určité nevýhody. Postfix nemá právo vytvárať priečinky, ani zapisovať do domovského adresára používateľa *root*, preto bol v jeho domovskom adresári vytvorený súbor **.forward** s obsahom *mic*hal@viphome.sk, ktorý všetku poшту pre používateľa *root* posieľa používateľovi *mic*hal. Pošta pre neexistujúcich užívateľov bola doručovaná užívateľovi *mic*hal (*luser_relay*). Posledným parametrom je *smtpd_banner*, ktorý sa zobrazí užívateľovi pri prihlásení na server SMTP cez telnet.

Po ukončení editovania súboru *main.cf* si overíme konfiguračný súbor príkazom *postfix check* a môžeme znovu nahrat' všetky konfiguračné zmeny príkazom *postfix reload*. Nakoniec celý server reštartujeme príkazom *service postfix restart* ako užívateľ *root*.

8.4 Konfigurácia autentifikácie SMTP

Pri konfigurácii autentifikácie bol využitý spomínaný systém Cyrus-SASL, ktorého knižnice sú odvodené od pôvodnej implementácie systému Cyrus.

8.4.1 Inštalácia Cyrus SASL

Systém Cyrus-SASL býva súčasťou väčšiny distribúcií. V tomto prípade bol do systému doinštalovaný príkazom:

```
[root@pluto /]# yum install cyrus-sasl cyrus-sasl-devel cyrus-sasl-gssapi cyrus-sasl-lib  
cyrus-sasl-md5 cyrus-sasl-plain
```

Po inštalácii systému Cyrus-SASL je potrebné si overiť, či je Postfix nainštalovaný s podporou Cyrus-SASL pomocou príkazu:

```
[root@pluto /]# postconf -a  
cyrus  
dovecot
```

Následne bol upravený konfiguračný súbor logovacieho démona *rsyslogd* tak, aby zaznamenával aj logovacie správy od systému Cyrus-SASL. V konfiguračnom súbore *rsyslog.conf* bol zmenený riadok so zápisom *auth, authpriv.** na:

- *auth.*, authpriv.** */var/log/auth.log*

Po tomto kroku bolo potrebné reštartovať logovacieho démona tak, ako v časti 7.2.3.

8.4.2 Konfiguračný súbor pre Postfix

„Aplikácie, ktoré ponúkajú služby SASL, musia vedieť, ako majú pracovať s knižnicami SASL. V systéme Cyrus SASL existuje pre každú aplikáciu zvláštny konfiguračný súbor, nie jeden súbor pre všetky. Pre rôzne aplikácie môžeme definovať rôzne konfiguračné súbory. Konfiguračný súbor Postfixu sa volá *smtpd.conf*, pretože v Postfixe je implicitnou aplikáciou, ktorá zaisťuje služby SASL démon *smtpd* [20].“

Konfiguračný súbor bol umiestnený v adresári */usr/lib/sasl2/*. Jeho obsah bol zmenený nasledovne:

```
log_level: 3  
pwcheck_method: auxprop  
mech_list: plain login cram-md5 digest-md5  
auxprop_plugin: sasl_db
```

Parameter *log_level* : 3 zabezpečí logovanie menej závažných varovaní. Parameter *pwcheck_method* hovorí Postfixu, ktorú službu overovania hesiel používateľov budeme využívať. Keďže chceme na serveri podporovať autentizačné procedúry CRAM-MD5 a DIGEST-MD5 bola vybraná autentizačná metóda *auxprop*, ktorá využíva na overovanie hesiel pomocné zásuvné moduly. Nasleduje výber overovacích mechanizmov SMTP

AUTH pomocou parametra *mech_list*. Posledný parameter *auxprop_plugin: sasl* definuje použitie štandardného zásuvného modulu systému Cyrus SASL. Tento modul má dve utility *saslpasswd2* a *sasldblistusers2*. Dvojky na konci príkazov označujú, že bola použitá verzia Cyrus-SASL 2.x. Keďže používame metódu *sasldb* ďalej už nebol potrebný démon *saslauthd*, ktorý komunikuje s autentizačnými procedúrami a bol v systéme vypnutý príkazmi:

```
[root@pluto /]# /etc/init.d/saslauthd stop  
[root@pluto /]# chkconfig --level 0123456 saslauthd off
```

8.4.3 Príprava Postfix na použitie SMTP AUTH

Uistíme sa, že v konfiguračnom súbore serveru Postfix *main.cf* sú pomocou znaku komentára na začiatku riadku vypnuté nasledovné voľby:

```
#mailbox_transport = lmtp:unix:/var/lib/imap/socket/lmtp  
#mailbox_transport = cyrus
```

Na koniec konfiguračného súboru *main.cf* boli doplnené konfiguračné parametre pre podporu SMTP AUTH:

```
smtpd_sasl_path = sasl2/smtpd  
smtpd_sasl_auth_enable = yes  
smtpd_sasl_security_options = noanonymous  
smtpd_sasl_local_domain = viphome.sk  
broken_sasl_auth_clients = yes
```

Prvým parametrom bola zapnutá samotná podpora SMTP AUTH. Druhý parameter prikazuje Postfixu, aby neponúkal klientom overovací mechanizmus ANONYMOUS, ktorý bol vytvorený kvôli anonymnému prístupu k poštovým službám. Postfix môže prostredníctvom parametra *smtpd_sasl_local_domain* pridať doménové meno k SASL prihlasovaciemu menu. Z užívateľa, ktorý má účet v systéme s menom *Michal* bude *Michal@viphome.sk*. Posledným parametrom prikazujeme Postfixu, aby podporoval aj starších e-mailových klientov ako sú Microsoft Outlook Express verzie 4 a Microsoft Exchange verzie 5.0.

Aby bolo umožnené posielanie pošty tým klientom, ktorí sa úspešne autentifikovali a ktorí patria do našej siete, do konfiguračného súboru boli doplnené nasledovné voľby:

```
smtpd_relay_restrictions =  
    permit_sasl_authenticated,  
    permit_mynetworks,  
    reject_unauth_destination
```

Po zmene v konfiguračnom súbore *main.cf* je potrebné znovu načítať konfiguráciu príkazom *postfix reload*, alebo reštartovať celý server príkazom *service postfix restart*. Pred týmto krokom si môžeme overiť svoju konfiguráciu príkazom *postfix check*. Úspešnosť konfigurácie si môžeme overiť tak, že sa prihlásime na server pomocou protokolu *telnet* na port 25 a vidíme dva tučne zvýraznené riadky:

```
220 pluto.viphome.sk ESMTP Postfix (2.6.6)  
EHLO michal.viphome.sk  
250-pluto.viphome.sk  
250-PIPELINING  
250-SIZE 10240000  
250-VRFY  
250-ETRN  
250-AUTH DIGEST-MD5 PLAIN CRAM-MD5 LOGIN  
250-AUTH=DIGEST-MD5 PLAIN CRAM-MD5 LOGIN  
250-ENHANCEDSTATUSCODES  
250-8BITMIME  
250 DSN
```

8.4.4 Databáza sasl_db2

Je veľmi dôležité v databáze *sasl_db2* vytvoriť užívateľov správne podľa toho, čo bolo nakonfigurované v prechádzajúcich častiach, inak autentizácia SMTP zlyhá. Užívateľov vytvárame pomocou spomenutého pluginu *saslpasswd2*, keďže používame systém Cyrus-SASL verzie 2.x. Záznam pre užívateľa s užívateľským účtom s menom *michal* v systéme bol vytvorený v databáze *sasl_db2* príkazom (zadáme aj platné heslo k užívateľskému účtu):

```
[root@pluto /]# saslpasswd2 -c -u viphome.sk michal  
Password:*****  
Again (for verification):*****
```

K tejto databáze bol obmedzený prístup iba pre užívateľa *root* a skupinu *postfix*:

```
[root@pluto /]# chmod 640 /etc/sasl2  
[root@pluto /]# chgrp postfix /etc/sasl2
```

Pomocou príkazu *sasldblistusers2* môžeme vypísať existujúcich užívateľov v databáze:

```
[root@pluto /]# sasldblistusers2  
eva@viphome.sk: userPassword  
michal@viphome.sk: userPassword
```

8.4.5 Test autentifikácie

V konfigurácii Postfixu v časti 8.3 bolo zápisom v parametri *mynetworks* povolené posielanie pošty zo siete 192.168.1.0/24. Keďže náš testovací počítač bol v rovnakej sieti, dočasne bol tento zápis vymazaný a v parametri *mynetworks* ponechaná len hodnota 127.0.0.0/8, ktorá spôsobila, že e-maily sa dali posilať len priamo zo samotného serveru. Po tejto zmene bolo potrebné server Postfix reštartovať, alebo nanovo načítať konfiguračný súbor.

Po úspešnej konfigurácii autentifikácie by však Postfix mal umožniť posielanie pošty aj klientom, ktorí sa úspešne autentifikujú aj keď sú v inej sieti.

Pomocou *telnet* sme sa prihlásili na server na porte 25 z počítača s IP adresou 192.168.1.100/24 a pokúsili sa odoslať testovaciu správu (výpis je skrátený len na potrebné príkazy):

```
220 pluto.viphome.sk ESMTP Postfix (2.6.6)  
EHLO michal.viphome.sk  
.....  
250-AUTH DIGEST-MD5 PLAIN CRAM-MD5 LOGIN  
250-AUTH=DIGEST-MD5 PLAIN CRAM-MD5 LOGIN  
.....  
MAIL FROM:<michal>  
250 2.1.0 Ok  
RCPT TO:<michal.hvizdak@centrum.sk>  
554 5.7.1 <michal.hvizdak@centrum.sk>: Recipient address rejected: Relay access denied
```

Server odmietol odoslať testovaciu správu, pretože klient nepochádza z dôveryhodnej siete a ani sa serveru nijako neautentifikoval.

Pomocou nasledovného príkazu boli zakódované prihlasovacie údaje do base64 bez znaku konca riadku:

```
[root@pluto /]# % echo -ne '\000michal\000heslo' | openssl base64
```

Zakódovaný reťazec bol vložený do *telnet* relácie a test zopakovaný:

```
220 pluto.viphome.sk ESMTP Postfix (2.6.6)
```

```
EHLO michal.viphome.sk
```

```
250-pluto.viphome.sk
```

```
.....
```

```
AUTH PLAIN AG1pY2hhbABtaWNoYWw=
```

```
235 2.7.0 Authentication successful
```

```
MAIL FROM:<michal>
```

```
250 2.1.0 Ok
```

```
RCPT TO:<michal.hvizdak@centrum.sk>
```

```
250 2.1.5 Ok
```

```
DATA
```

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

```
Test autentizacie klienta po konfigurácii SMTP AUTH.
```

```
.
```

```
250 2.0.0 Ok: queued as 5D0D21FB4B
```

Posledný riadok je odpoveďou servera na ukončenie našej e-mailovej správy. Server zaradil správu do fronty správ na odoslanie. Týmto bola úspešne overená autentifikácia klientov.

8.5 Vytvorenie e-mailového účtu

Pre úspešné vytvorenie e-mailového účtu bolo potrebné najskôr vytvoriť pre daného užívateľa systémový účet. Predpokladajme, že sme vytvorili užívateľský účet s menom *jozef* a heslom *password123*. Po tomto kroku je e-mailový účet plne funkčný a môžeme z neho posielat' poštu v rámci sietí, z ktorých je povolené odosielenie pošty. Chceme však, aby užívateľ mohol posielat' poštu aj z inej siete a používať tak náš server napr. z domu. Aby sme užívateľovi sprístupnili aj túto funkciu servera bolo potrebné vytvoriť záznam s jeho menom v autentizačnej databáze a zadať platné heslo k lokálnemu účtu:

```
[root@pluto /]# saslpaswd2 -c -u viphome.sk jozef
```

9 INŠTALÁCIA ANTIVÍRUS A ANTISPAM

Ako antivírus bol v tejto práci použitý open-source softvér ClamAV, ktorý je určený na detekciu trójskych koní, vírusov, malware a iných hrozieb. Je štandardným antivírusovým programom používaným na e-mailových bránach. Spolu s týmto programom bol použitý open-source softvér MailScanner a ako antispamové riešenie využitý program SpamAssassin. Pred inštaláciou týchto softvérov bolo potrebné do systému nainštalovať repozitáre EPEL (Extra Packages for Enterprise Linux), ktorých inštalácia je popísaná ďalej.

9.1 Pridanie repozitárov EPEL

Repozitár je typom špeciálneho servera na internete odkiaľ môžeme sťahovať balíčky s aktualizáciami programov, systémovými aktualizáciami alebo z neho môžeme priamo inštalovať aplikácie, ktoré v systéme ešte nemáme. Repozitáre EPEL spolu s balíčkom *yum-priorities* boli nainštalované príkazmi:

```
[root@pluto ~]# rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
```

```
[root@pluto ~]# yum install yum-priorities
```

Do súboru */etc/yum.repos.d/epel.repo* pod odsek *[epel]* bol pridaný parameter *priority=10*. Úspešnosť inštalácie bola overená príkazom *yum repolist*. Vo výstupe príkazu by sme mali vidieť riadok začínajúci slovom *epel*:

```
[root@pluto ~]# yum repolist
```

```
Loaded plugins: refresh-packagekit, security
```

<i>repo id</i>	<i>repo name</i>	<i>status</i>
<i>epel</i>	<i>Extra Packages for Enterprise Linux 6 - x86_64</i>	<i>8,583</i>

9.2 Inštalácia ClamAV

Softvér ClamAV bol nainštalovaný jednoducho príkazom *yum install clamd*. ClamAV poskytuje nástroje na skenovanie súborov priamo z príkazového riadku. Môžeme využiť príkaz *clamscan*, ktorý bez použitia parametrov prezrie súčasný pracovný adresár, v ktorom sa nachádzame. Príkazom *freshclam* bola aktualizovaná vírusová databáza programu. Je zrejmé, že antivírusový program by mal mať aktuálnu databázu vírusových hrozieb, preto

musíme zabezpečiť jej pravidelnú aktualizáciu. Na túto činnosť bola využitá pomocná súčasť antivírusu ClamAV – *freshclam*, ktorá môže pracovať v dvoch módoch. Priamo z príkazového riadku na požiadanie, alebo ako démon na pozadí systému. Aby sme mohli zaznamenávať činnosť tejto súčasti bol vytvorený logovací súbor:

```
[root@pluto]# touch /var/log/freshclam.log  
[root@pluto]# chmod 600 /var/log/freshclam.log  
[root@pluto]# chown clam /var/log/freshclam.log
```

Do konfiguračného súboru */etc/freshclam.conf* k parametru **UpdateLogFile** bola pridaná cesta k súboru, do ktorého sa má logovať činnosť démona, ktorý bude aktualizovať antivírus ClamAV. Následne bol spustený pomocný program *freshclam*, ako démon príkazom *freshclam -d*. Spúšťanie démona po každom prípadnom reštarte systému bolo zabezpečené pridaním zápisu "*freshclam -d*" do skriptu */etc/rc.local*.

9.3 Inštalácia MailScanner a SpamAssassin

MailScanner je antivírus a antispam pre e-mailové servery. Je vytvorený pre použitie na Linuxových e-mailových bránach, aby užívatelia a ich e-maily mohli byť chránení z jedného miesta. Pred samotnou inštaláciou bolo potrebné nainštalovať najskôr všetky potrebné závislosti programu MailScanner príkazom:

```
[root@pluto]# yum install perl-Archive-Zip perl-DBI perl-DBD-SQLite perl-Filesys-Df  
perl-Net-CIDR perl-OLE-Storage_Lite perl-Sys-Hostname-Long perl-Sys-SigAction  
perl-MIME-tools perl-Time-HiRes perl-HTML-Parser perl-Compress-Zlib
```

Zo webovej stránky produktu MailScanner (<http://www.mailscanner.info/>) bola stiahnutá najnovšia stabilná verzia programu pre systémy Red Hat a uložená do adresára */install*. MailScanner bol nainštalovaný postupne týmito príkazmi:

```
[root@pluto install]# gunzip MailScanner-4.84.5-3.rpm.tar.gz  
[root@pluto install]# tar -xvf MailScanner-4.84.5-3.rpm.tar  
[root@pluto install]# yum install --nogpgcheck mailscanner* tnef*
```

Softvér SpamAssassin bol nainštalovaný do systému príkazom:

```
[root@pluto /]# yum install spamassassin
```

9.4 Konfigurácia MailScanner a SpamAssassin

Konfigurácia softvéru MailScanner bola vykonaná úpravou konfiguračných súborov. V konfiguračnom súbore *virus.scanners.conf* v adresári */etc/MailScanner/* sme vyhľadali riadok začínajúci textom *clamav* a jeho cesta k programu ClamAV antivírus bola zmenená na */usr/share/clamav*. V konfiguračnom súbore *MailScanner.conf* v adresári */etc/MailScanner/* boli zmenené tieto riadky:

```
%org-name% = VIPHOME
%org-long-name% = Viphome s.r.o.
%web-site% = www.viphome.sk
Run As User = postfix
Run As Group = postfix
Incoming Queue Dir = /var/spool/postfix/hold
Outgoing Queue Dir = /var/spool/postfix/incoming
MTA = postfix
Incoming Work Group = clam
Incoming Work Permissions = 0640
Virus Scanners = clamd
Clamd Socket = /var/run/clamav/clamd.sock
Use SpamAssassin = yes
SpamAssassin User State Dir = /var/spool/MailScanner/spamassassin
```

Na koniec konfiguračného súboru bol doplnený parameter **High Scoring Spam Actions = deliver**. Tento parameter zabezpečí, že užívateľovi bude doručený aj prípadný spam a užívateľ sám vo svojej schránke rozhodne, či sa o spam jedná alebo nie.

V konfiguračnom súbore */etc/MailScanner/spam.assassin.prefs.conf* v parametri *bayes_ignore_header* bol zmenený reťazec *YOURDOMAIN* medzi *X-* a *-MailScanner* na hodnotu, ktorá bola nastavená vyššie do parametra *%org-name%*. V tomto prípade reťazec *VIPHOME*. Zmena bola vykonaná vo všetkých štyroch parametroch. Do parametra *envelope_sender_header* medzi *X-* a *-MailScanner* bola takisto doplnená hodnota z *%org-name%*. Parameter *use_auto_whitelist 0* bol vypnutý pomocou znaku komentára „#” na začiatku riadku.

V adresári */var/spool/MailScanner/* bol vytvorený adresár *spamassassin* a práva zmenené nasledovne:

```
[root@pluto install]# cd /var/spool/MailScanner/  
[root@pluto MailScanner]# mkdir spamassassin  
[root@pluto MailScanner]# chown -R postfix.clam *  
[root@pluto MailScanner]# chmod -R 750 *  
[root@pluto MailScanner]# restorecon -R ../MailScanner/
```

Ďalej bol nakonfigurovaný Postfix tak, aby všetky prijaté správy najskôr poslal to fronty *hold* až kým správy neskontroluje MailScanner. V súbore */etc/postfix/main.cf* bol povolený parameter *header_checks = regexp:/etc/postfix/header_checks*. Na koniec súboru */etc/postfix/header_checks* bol doplnený príkaz *^Received:/ HOLD*. Po tejto konfigurácii bude program MailScanner zodpovedný za spúšťanie Postfixu. Vykonané boli tieto príkazy v tomto poradí (poradie príkazov je dôležité):

```
[root@pluto /]# chkconfig postfix off  
[root@pluto /]# chkconfig --levels 2345 postfix off  
[root@pluto /]# chkconfig clamd on  
[root@pluto /]# chkconfig MailScanner on  
[root@pluto /]# service postfix stop  
[root@pluto /]# service clamd start
```

Pri vykonaní posledného príkazu bolo veľmi dôležité, aby softvér MailScanner sám spustil MTA Postfix:

```
[root@pluto /]# service MailScanner start
```

Starting MailScanner daemons:

<i>incoming postfix:</i>	<i>[OK]</i>
<i>outgoing postfix:</i>	<i>[OK]</i>
<i>MailScanner:</i>	<i>[OK]</i>

Úspešnosť konfigurácie a správny štart servera Postfix bol overený pomocou relácie *telnet*:

```
[root@pluto /]# telnet localhost smtp
```

```
220 pluto.viphome.sk ESMTP Postfix (2.6.6)
```

```
quit
```

```
221 2.0.0 Bye
```

```
Connection closed by foreign host.
```


9.5 Test nastavenia

Konfigurácia MailScanner a SpamAssassin bola overená z príkazového riadku príkazom *MailScanner --lint*. Výpis príkazu bol bez chýb. Test antivíru a spamového filtra bol overený odoslaním testovacej vírusovej správy a testovacej spamovej správy. Najskôr bol nainštalovaný e-mailový klient *mutt*, ktorého môžeme použiť z príkazového riadku. Následne bola odoslaná testovacia vírusová a spamová správa pre používateľa *root*. Obe správy boli zachytené a označené systémom MailScanner:

```
[root@pluto /]# yum install mutt
```

```
[root@pluto /]# cd /usr/share/doc/clamav-0.97.3/test/
```

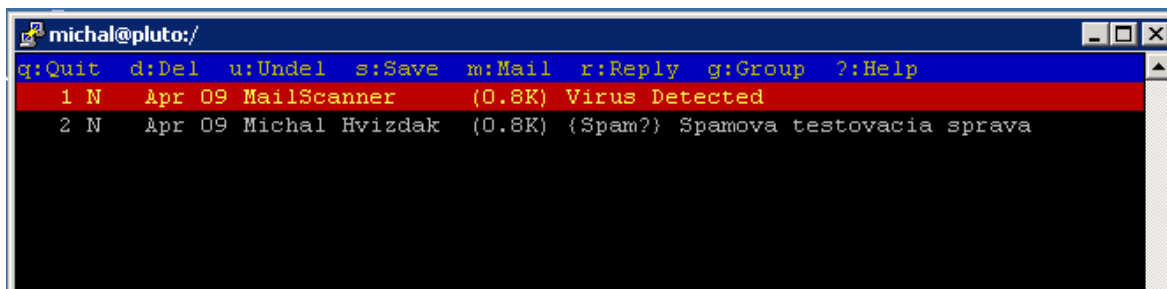
```
[root@pluto test]# cat clam.mail | mail -s "Virusova testovacia sprava" root
```

```
[root@pluto test]# cd /usr/share/doc/spamassassin-3.3.1/
```

```
[root@pluto spamassassin-3.3.1]# cat sample-spam.txt | mail -s "Spamova testovacia sprava" root
```

```
[root@pluto /]# mutt
```

Po spustení klienta *mutt* boli obe poslané správy detekované systémom MailScanner ako vírusová a spamová správa (Obr. 12). Je dobré poznamenať, že v časti 8.3 bolo nastavené presmerovanie e-mailov používateľa *root* pre používateľa *Michal*.



Obr. 12 – Detekcia vírusu a spamu v e-mailových správach.

10 KONFIGURÁCIA PRÍJÍMU E-MAILOV PRE DOMÉNU

U poskytovateľa internetových služieb bol zriadený doménový kôš, t.j. e-mailová schránka, ktorá prijíma e-maily pre celú doménu nezávisle od skutočného adresáta. V časti 4.2 sme hovorili o dvoch častiach e-mailovej adresy. Miestna časť LHS a doména RHS. V prípade doménového koša je teda smerodajná informácia RHS, teda doména, do ktorej sa má konkrétny e-mail doručiť. Pre náš e-mailový server bude smerodajná informácia LHS, vďaka ktorej bude vedieť, ktorému skutočnému používateľovi na našom serveri sa má e-mail doručiť.

Na prenos e-mailov od poskytovateľa na náš poštový server bol použitý program Fetchmail, ktorého autorom je Eric S. Raymond. Je určený na diaľkové získavanie e-mailov prostredníctvom protokolov POP a IMAP. Program sa spojí s požadovanou poštovou schránkou, nájdenú poštu prevezme a odovzdá programu MTA. V našom prípade poštu predá serveru Postfix, ktorý ju roztriedi do užívateľských schránok. Konfiguračný súbor programu Fetchmail sa volá **.fetchmailrc** (dôležitá je bodka na začiatku) a je umiestnený v domovskom adresári užívateľov.

Bol vytvorený systémový účet, ktorý neumožňuje prihlásenie, ale má domovský adresár a bude slúžiť len na výber pošty pre všetkých užívateľov nášho servera, zo vzdialeného servera. Účet bol vytvorený pomocou príkazu *useradd* s prepínačom *-r*:

```
[root@pluto michal]# useradd fetchmail -r -m
```

Program Fetchmail bol nainštalovaný príkazom *yum install fetchmail*. V domovskom adresári používateľa *fetchmail* bol vytvorený spomínaný súbor **.fetchmailrc**. Vlastník súboru bol zmenený na používateľa *fetchmail* a skupinu na *fetchmail*. Prístupové práva súboru boli nastavené tak, aby súbor mohol čítať a zapisovať do neho len vlastník:

```
[fetchmail@pluto ~]$ ls -la .fetchmailrc  
-rw----- 1 fetchmail fetchmail 1372 Apr 10 10:29 .fetchmailrc
```

Obsah súboru **.fetchmailrc** je nasledovný:

```
set syslog  
set postmaster michal  
defaults timeout 300,  
    nokeep,
```

poll pop3.viphome.sk aka viphome.sk no dns no envelope

protocol POP3

user "all@viphome.sk"

there with password "HESLO"

*to * here*

smtp host localhost

Prvým parametrom *set syslog* bolo nastavené zaznamenávanie činnosti programu Fetchmail do systémového logu, ktorý bol nakonfigurovaný v časti 7.2.2. Parameter *set postmaster* spôsobí, že e-maily pre neexistujúceho lokálneho používateľa budú doručené používateľovi *Michal*. Interval sťahovania pošty bol nastavený na päť minút a kópie správ nechceme ponechávať na serveri poskytovateľa (*timeout 300, nokeep*).

Najdôležitejší je riadok začínajúci príkazom *poll*. Kontaktujeme server *pop3.viphome.sk* pre doménu *viphome.sk* pomocou protokolu *POP3*. Pošta bola prenášaná zo schránky *all@viphome.sk*, ktorá je u poskytovateľa doménovým košom a sťahovaná pre ľubovoľného používateľa lokálneho systému – *to * here*. Posledný riadok definuje, ktorému serveru má byť pošta po stiahnutí odovzdaná. Server poskytovateľa bol kontaktovaný priamo zo servera, na ktorom máme spustený MTA Postfix zadáme preto *smtp host localhost*.

Spúšťanie programu Fetchmail, ktorý bude hľadať tento konfiguračný skript bolo zabezpečené pomocou démona *cron*. Bol vytvorený crontab pre užívateľa *fetchmail*, s obsahom, ktorý zabezpečí pravidelné spúšťanie v intervaloch 5 minút:

```
*/5 * * * * fetchmail > /dev/null 2>&1
```

Týmto bol vyriešený príjem pošty, ktorá je poslaná pre užívateľov, ktorí na našom serveri existujú. Pošta pre neexistujúcich používateľov zatiaľ doručená nebude a vráti sa odosielateľovi ako nedoručiteľná napriek tomu, že by mala byť doručená používateľovi *Michal*. Tento problém bol vyriešený úpravou konfiguračného súboru *main.cf* servera Postfix. Parameter *user_relay* bol doplnený o lokálneho používateľa, ktorý bude dostávať zvyšnú poštu z domény *viphome.sk*. Parameter *local_recipient_map* bol ponechaný prázdny bez komentáru na začiatku riadku. Server Postfix bolo potrebné reštartovať, aby sa zmeny v konfigurácii uplatnili.

11 INŠTALÁCIA A KONFIGURÁCIA DOVECOT

11.1 Inštalácia Dovecot

Inštalácia POP/IMAP servera Dovecot bola v systéme Red Hat veľmi jednoduchá. Boli použité nasledovné príkazy, ktorými bol server nainštalovaný a zabezpečené jeho spúšťanie po štarte systému:

```
[root@pluto /]# yum install dovecot  
[root@pluto /]# service dovecot start  
[root@pluto /]# chkconfig --level 2345 dovecot on
```

11.2 Konfigurácia Dovecot

Konfigurácia POP/IMAP servera Dovecot bola jednoduchá, hlavne kvôli faktu, že po inštalácii servera Dovecot sú všetky voľby pripravené a bolo potrebné zapísať do konfiguračných súborov len také hodnoty, ktoré sa zhodujú s prostredím, kde server Dovecot plánujeme používať. Umiestnenie konfiguračných súborov bolo zistené pomocou príkazu:

```
[root@pluto /]# doveconf -n | head -1
```

Prvou veľmi dôležitou vecou, ktorú bolo potrebné zapnúť je podpora protokolov, ktoré chceme používať na prijímanie používateľskej pošty. V súbore */etc/dovecot/dovecot.conf* do parametra *protocols* bola pridaná podpora protokolov POP a IMAP (*protocols = pop3 imap*). Parameter *mail_location* v konfiguračnom súbore */etc/dovecot/conf.d/10-mail.conf* bol zmenený na hodnotu "*maildir: ~/Maildir*", pretože e-mailové schránky používateľov sú v tejto práci umiestnené priamo v domovských adresároch užívateľov vo formáte Maildir. Parameter *pop3_uidl_format* v konfiguračnom súbore */etc/dovecot/conf.d/20-pop3.conf* bol zmenený na hodnotu "*%08Xu%08Xv*" a parameter *pop3_clients_workarounds* na "*outlook-no-nuls oe-ns-eoh*". Prvým parametrom bolo nadefinované aký UIDL (Unique Mail Identifier) má server Dovecot použiť. Pomocou druhého parametra bola zabezpečená kompatibilita so staršími e-mailovými klientmi ako sú Outlook, Outlook Express alebo Netscape Mail. Bolo potrebné reštartovať server Dovecot aby sa zmeny v konfigurácii uplatnili a jeho funkčnosť bola overená prihlásením sa k poštovej schránke pomocou najskôr pomocou protokolu POP:

[michal@pluto /]\$ telnet localhost pop3

Connected to localhost.

Escape character is '^]'.

+OK Dovecot ready.

user michal

+OK

pass ***

+OK Logged in.

list

+OK 2 messages:

1 440

2 2536

a následne pomocou protokolu IMAP:

[michal@pluto /]\$ telnet localhost imap

Connected to localhost.

Escape character is '^]'.

** OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE STARTTLS AUTH=PLAIN] Dovecot ready.*

1 login michal ***

1 OK Logged in

2 list "" ""

** LIST (\HasNoChildren) "." "INBOX"*

2 OK List completed.

5 select INBOX

** FLAGS (\Answered \Flagged \Deleted \Seen \Draft)*

** OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft *)] Flags permitted.*

** 9 EXISTS*

** 0 RECENT*

** OK [UIDVALIDITY 1365336601] UIDs valid*

** OK [UIDNEXT 10] Predicted next UID*

** OK [HIGHESTMODSEQ 1] Highest*

5 OK [READ-WRITE] Select completed.

12 INŠTALÁCIA A KONFIGURÁCIA WEBMAIL

Webmail je e-mailový klient implementovaný ako webová aplikácia, ku ktorej môžeme pristupovať cez ľubovoľný webový prehliadač. V tejto práci bol použitý open-source softvér SquirrelMail, ktorý je napísaný v jazyku PHP (Hypertext Preprocessor). SquirrelMail potrebuje k svojej činnosti webový server, SMTP server a POP/IMAP server. Využitý bol webový server Apache, SMTP server Postfix a POP/IMAP server Dovecot.

12.1 Inštalácia servera Apache a PHP

Po úspešnej inštalácii repozitárov EPEL a REMI, ktorých inštalácia bola popísaná v časti 9.1 bola inštalácia servera Apache a potrebných balíčkov PHP veľmi jednoduchá a vykonaná nasledujúcimi príkazmi:

```
[root@pluto /]# yum install httpd php php-common php-mysql php-pgsql php-sqlite php-xml
```

Po inštalácii bol server spustený a ďalej bolo nakonfigurované jeho spúšťanie v systémových úrovniach behu 2,3,4 a 5:

```
[root@pluto /]# service httpd start
```

```
[root@pluto /]# chkconfig --level 2345 httpd on
```

Úspešnosť inštalácie PHP bola overená pomocou súboru *test.php* s nasledovným obsahom:

```
<?php
    phpinfo();
?>
```

Tento súbor bol vytvorený v adresári */var/www/html/*. Do webového prehliadača zapíšeme adresu *http://localhost/test.php*. V prehliadači by sa mali zobrazit' informácie o nainštalovanom jazyku PHP.

12.2 Inštalácia a konfigurácia SquirrelMail

Webmail SquirrelMail bol do systému nainštalovaný jednoducho príkazom *yum install squirrelmail*.

Konfigurácia SquirrelMail bola vykonaná prostredníctvom skriptu jazyka Perl. Konfiguračné rozhranie vidíme na obrázku (Obr. 13).

Do príkazového riadku zadáme príkaz:

```
[root@pluto /]# /usr/share/squirrelmail/config/config.pl
```

Vyberieme voľbu "**D**", pomocou ktorej boli nastavené preddefinované hodnoty pre POP/IMAP servery. Keďže používame POP/IMAP server Dovecot napíšeme do sprievodcu *dovecot* a stlačíme „enter“.

V časti "**1. Organization Preferences**", bolo nakonfigurované meno organizácie, ktorá bude používať webmail a všetky potrebné informácie.

V časti "**2. Server Settings**" vyberieme voľbu "**1. Domain**" a napíšeme našu doménu – *viphome.sk*.. Ďalej vyberieme voľbu "**3. Sendmail or SMTP**" a v podmenu vyberieme "**SMTP**" tak, ako vidíme na obrázku (Obr. 14). Nastavenia uložíme príkazom "**S**" a vrátime sa späť do hlavného menu. Po tomto kroku môžeme sprievodcu ukončiť.

Týmto boli nakonfigurované hlavné časti webmailu, ktorý môžeme začať používať.



```
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Main Menu --
1.  Organization Preferences
2.  Server Settings
3.  Folder Defaults
4.  General Options
5.  Themes
6.  Address Books
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database
10. Languages

D.  Set pre-defined settings for specific IMAP servers

C   Turn color off
S   Save data
Q   Quit

Command >> █
```

Obr. 13 – Konfiguračné rozhranie programu SquirrelMail.

```
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Server Settings

General
-----
1. Domain           : viphome.sk
2. Invert Time       : false
3. Sendmail or SMTP  : SMTP

A. Update IMAP Settings : localhost:143 (dovecot)
B. Update SMTP Settings : localhost:25
```

Obr. 14 – Konfigurácia domény, SMTP a IMAP pre webmail.

Pre správne fungovanie webmailu bolo nutné ešte upraviť konfiguračný skript webového servera Apache **httpd.conf** v adresári `/etc/httpd/conf/`. Na koniec bola pridaná nasledovná konfigurácia:

```
Alias /squirrelmail /usr/share/squirrelmail
```

```
<Directory /usr/share/squirrelmail>
```

```
Options Indexes FollowSymLinks
```

```
RewriteEngine On
```

```
AllowOverride All
```

```
DirectoryIndex index.php
```

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

Následne bolo potrebné reštartovať webový server Apache. Správna funkcionálnosť webmailu bola overená na adrese `http://192.168.1.1/webmail` (Obr. 15).



Obr. 15 – Prihlasovací dialóg programu SquirrelMail.

12.3 Inštalácia doplnkov SquirrelMail

12.3.1 Inštalácia Retrieve User Data plugin

Plugin *Retrieve User Data* dokáže vyčítať celé užívateľské meno a e-mailovú adresu z externého zdroja a zapísať tieto hodnoty do užívateľských nastavení webmailového prostredia. Plugin bol nainštalovaný pomocou nasledujúcich príkazov:

```
[root@pluto install]# wget http://www.linuxmail.info/files/retrieveuserdata.0.9-1.4.3.tar.gz
[root@pluto install]# gunzip retrieveuserdata.0.9-1.4.3.tar.gz
[root@pluto install]# tar -xvf retrieveuserdata.0.9-1.4.3.tar
[root@pluto install]# cp -r retrieveuserdata /usr/share/squirrelmail/plugins/
```

Ďalej bolo potrebné z terminálu spustiť konfiguračný skript programu SquirrelMail */usr/share/squirrelmail/config/conf.pl* a vybrať voľbu "**8. Plugins**". Voľbou čísla, ktoré je pridelené pluginu *retrieveuserdata* bol plugin povolený. V hlavnom menu v časti "**4. General Options**" a podmenu "**9. Allow editing of identity**" boli zmenené všetky voľby na "**false**". Tým zabránime užívateľom editovať svoje meno, priezvisko a e-mail. Tieto nastavenia budú vyplnené automaticky.

12.3.2 Inštalácia Autocomplete plugin

Plugin *Autocomplete* hľadá užívateľov v užívateľových kontaktoch počas písania do polí To/CC/Bcc pri písaní novej správy v prostredí webmailu a dopĺňa relevantné výsledky. Jeho inštalácia bola vykonaná obdobne spôsobom popísaným vyššie:

```
[root@pluto install]# wget
http://squirrelmail.org/countdl.php?fileurl=http%3A%2F%2Fwww.squirrelmail.org%2Fplugins%2Fautocomplete-3.0-1.4.0.tar.gz
[root@pluto install]# gunzip autocomplete-3.0-1.4.0.tar.gz
[root@pluto install]# tar -xvf autocomplete-3.0-1.4.0.tar
[root@pluto install]# cp -r retrieveuserdata /usr/share/squirrelmail/plugins/
```

Voľbou čísla, ktoré je pridelené pluginu *autocomplete* v časti "**8. Plugins**" bol plugin povolený.

13 INŠTALÁCIA A KONFIGURÁCIA FIREWALL

13.1 Inštalácia firewallu

Najznámejším a najpoužívanejším firewallom v systéme Linux je program Netfilter. Väčšinou je súčasťou distribúcie a Red Hat nie je výnimkou, ale ak to tak nie je, môžeme ho do systému doinštalovať.

V tejto bol vytvorený skript, pomocou ktorého budeme môcť zastavovať a spúšťať firewall na serveri podľa potreby. Všetky pravidlá programu Netfilter boli nadefinované do skriptu *firewall.sh* a tento bol umiestnený do adresára */etc/sysconfig/*.

13.2 Základné nastavenie systému

Systém Linux má množstvo zabudovaných ochranných mechanizmov, ktoré nám umožňujú zvýšiť bezpečnosť servera a mali by sme ich aktivovať. Všetky tieto parametre sú parametre jadra a môžeme ich modifikovať editáciou súboru *sysctl.conf* v adresári */etc/*. V tomto konfiguračnom boli upravené nasledovné hodnoty:

<i>net.ipv4.ip_forward = 1</i>	<i>#Podpora routovania packetov</i>
<i>net.ipv4.conf.default.rp_filter = 1</i>	<i>#Chrani pred IP spoofing</i>
<i>net.ipv4.conf.default.log_martians = 1</i>	<i>#Logovanie packetov zo "zlou" IP</i>
<i>net.ipv4.tcp_syncookies = 1</i>	<i>#Ochrana pred DOS útokmi</i>

13.3 Skript firewallu

Samotný skript sa skladá z viacerých častí. Na začiatku boli definované dve premenné *prog* a *VERSION*, v ktorých sa nachádza meno skriptu a jeho verzia, ktorá sa vypíše na terminál pri štarte skriptu. Ostatné premenné a príkazy skriptu boli nadefinované v častiach *start*, *stop* a *restart*. Najdôležitejšia je časť *start*, ktorej úlohou je celý skript spustiť a nastaviť potrebné pravidlá do programu Netfilter. V parametri *stop* sa vypnú všetky nastavené pravidlá, vymažú sa definované reťaze a nastaví sa štandardná politika – povoliť všetko. Parameter *restart* len znovu spustí časti *stop* a *start*.

Na začiatku skriptu v časti *start* boli v štyroch premenných nadefinované TCP a UDP (User Datagram Protocol) služby povolené pre lokálne LAN a verejné WAN rozhranie. Ďalej boli vytvorené užívateľské reťaze *local_lan* pre packety pochádzajúce z internej siete

a *internet_wan* pre ostatné packety. Vytvorené boli aj reťaze *valid-src* a *valid-dst*, ktoré budú slúžiť na overenie zdrojovej a cieľovej adresy zariadenia, ktoré sa snaží komunikovať s verejným sieťovým rozhraním firewallu. V ďalšej časti bolo nastavené maskovanie všetkých packetov z neverejnej siete LAN na verejnú IP adresu servera. Povolený bol ICMP (Internet Control Message Protocol) protokol z LAN siete na vnútorné LAN rozhranie aj na vonkajšie WAN rozhranie. Ping zo siete Internet zakážeme na obe sieťové rozhrania. Všetky neplatné packety server zahadzuje a vstup bol povolený len packetom, ktoré patria k už vytvoreným spojeniam, ktoré boli inicializované priamo z našej siete. Pomocou cyklu *for* boli povolené jednotlivé služby TCP a UDP definované v premenných na začiatku skriptu pre LAN a WAN rozhranie servera.

Samotný skript bol rozdelený do viacerých sekcií, ktoré sú v skripte označené komentárom a názvom príslušnej sekcie kvôli ľahšej orientácii. Skript obsahuje definíciu sietí, definíciu povolených služieb pre LAN a pre WAN, inicializáciu firewallu, kontrolu privátnych adries, sekciu maskovania packetov, delenie packetov a najdôležitejšiu časť povoľovanie služieb TCP a UDP pre rozhranie LAN a pre rozhranie WAN.

Skriptu bol umiestnený do adresára */etc/sysconfig/* a práva k súboru zmenené nasledovne:

```
[root@pluto /]# chmod 744 /etc/sysconfig/firewall.sh
```

Celý firewall ovládame pomocou príkazu */etc/sysconfig/firewall.sh* a pridáme *start*, *stop* alebo *restart* podľa operácie, ktorú chceme vykonať.

Po úspešnej aktivácii firewallu uvidíme v príkazovom riadku počet povolených TCP a UDP služieb pre jednotlivé rozhrania serveru a príkazom *iptables -L* si môžeme overiť nastavenia programu Netfilter.

Spúšťanie skriptu po štarte servera bolo zabezpečené pomocou symbolických odkazov na skript firewallu z jednotlivých adresárov pre úrovně behu systému:

```
[root@pluto /]# ln -s /etc/sysconfig/firewall.sh /etc/rc.d/rc0.d/K98firewall.sh
```

```
[root@pluto /]# ln -s /etc/sysconfig/firewall.sh /etc/rc.d/rc6.d/K98firewall.sh
```

```
[root@pluto /]# ln -s /etc/sysconfig/firewall.sh /etc/rc.d/rc2.d/S98firewall.sh
```

```
[root@pluto /]# ln -s /etc/sysconfig/firewall.sh /etc/rc.d/rc3.d/S98firewall.sh
```

```
[root@pluto /]# ln -s /etc/sysconfig/firewall.sh /etc/rc.d/rc5.d/S98firewall.sh
```

14 NÁSTROJE NA GRAFICKÚ SPRÁVU SERVERA

Najrozšírenejším nástrojom na grafickú správu servera priamo z webového prehliadača je program Webmin, ktorého autorom je Jamie Cameron a spol.. Projekt vznikol v roku 1997 a stal sa z neho veľmi úspešný nástroj na správu serverov, či už vzdialene alebo lokálne. Webmin je napísaný v jazyku Perl, čo zaručuje možnosť použitia na rôznych operačných systémoch od Windows po Linux. Skladá sa z vlastného webového servera, ktorý počúva na porte 10 000 a modulov, z ktorých každý vykonáva nejakú špecifickú činnosť na správu servera.

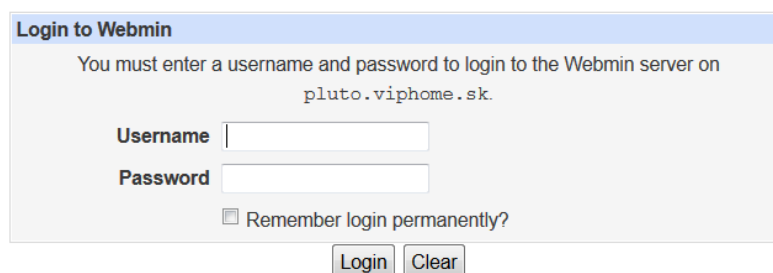
14.1 Inštalácia programu Webmin

Inštalácia programu v systéme Red Hat bola veľmi jednoduchá. Program bol nainštalovaný najjednoduchším spôsobom priamo z balíčku RPM pre tento systém týmito príkazmi:

```
[root@pluto install]# wget http://prdownloads.sourceforge.net/webadmin/webmin-1.620-1.noarch.rpm
```

```
[root@pluto install]# rpm -ivh webmin-1.620-1.noarch.rpm
```

Po úspešnej inštalácii bola overená funkcionálnosť programu Webmin na adrese `http://192.168.1.1:10000` (Obr. 1). Keďže bol program Webmin nainštalovaný z balíčka RPM prihlasovacie údaje sú podľa dokumentácie login a heslo používateľa `root`.



Obr. 16 – Prihlasovací dialóg programu Webmin.

Aktuálna verzia programu Webmin v čase písania tejto práce (v. 1.620) podporuje pripojenie k programu zabezpečeným pripojením pomocou SSL. Nebolo nutné inštalovať doplnkové knižnice SSL tak, ako tomu bolo v starších verziách programu.

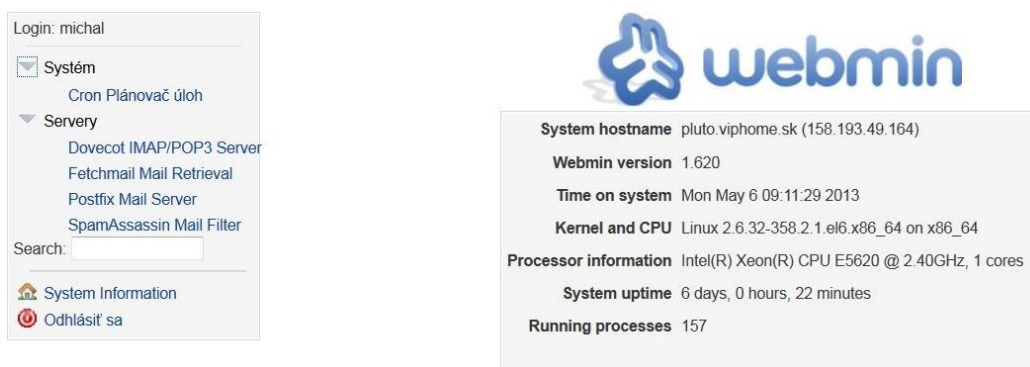
14.2 Základné nastavenia

Z bezpečnostného hľadiska je nežiadúce, aby k spravovanému serveru mohol pristupovať hocikto, či už z našej siete alebo z internetu. Boli vykonané základné nastavenia programu. V časti *Webmin – Konfigurácia webminu* boli pomocou modulu *Riadenie prístupu cez IP* nakonfigurované dôveryhodné adresy, z ktorých bol povolený prístup k programu. V našom prípade počítať administrátora s adresou 192.168.1.100. V module *Záznamy* bolo nastavené logovanie programu do systémových logov serveru. V časti *SSL kódovanie* bola zapnutá podpora kódovania SSL (Secure Socket Layer).

Prihlasovanie sa do programu pomocou užívateľského účtu používateľa *root* tak isto predstavuje bezpečnostné riziko. V časti *Webmin – Užívatelia Webminu* bola preto vytvorená skupina *administrators* a pridaný nový užívateľia. Na užívateľov programu je možné jednoduchým spôsobom prekonvertovať vybraných užívateľov, ktorí už reálne existujú na našom e-mailovom serveri. Po tomto kroku boli nastavené práva pre jednotlivých užívateľov. V časti *Užívatelia Webminu* vyberieme užívateľa a priradíme mu dostupné moduly z časti *Available Webmin Modules*. Pre užívateľa *michal* boli vybrané všetky moduly súvisiace s e-mailovým serverom (Dovecot POP/IMAP server, Fetchmail Mail Retrieval, SpamAssassin Mail Filter, Postfix Mail Server) a nastavenia uložené.

Po prihlásení do programu ako užívateľ *michal* boli k dispozícii len tie moduly, ktoré boli nastavené vyššie (Obr. 17). Užívateľ *michal* má práva spravovať všetky tieto časti e-mailového serveru tak ako používateľ *root*.

Týmto spôsobom je možné vytvoriť rôzne účty pre rôznych správcov nášho servera, pričom každý sa môže starať o svoju časť servera a prispôbíme mu k tomu aj rozhranie programu Webmin.



Obr. 17 - Užívateľské konfiguračné rozhranie programu Webmin.

ZÁVER

V tejto práci bola popisovaná tvorba kompletného e-mailového servera, od inštalácie systému Red Hat, konfiguráciu serverov SMTP, POP, IMAP až po vytvorenie firewallového skriptu. Inštalácia systému Red Hat bola jednoduchá a intuitívna. Najťažšia časť inštalácia bola konfigurácia RAID partícií a vytvorenie LVM logických zväzkov. Počas inštalácie nebol využitý celý priestor na pevných diskoch servera hlavne kvôli neskoršej možnosti rozšíriť existujúce partície.

Ako prvé po inštalácii bolo nastavené sieťové pripojenie. Na serveri je veľmi nežiaduce prevádzkovať rôzne všeobecne známe a nepotrebné služby, ktoré boli preto vypnuté. Systémová logovacia služba syslog bola nakonfigurovaná na oddelené logovanie jednotlivých e-mailových a autentifikačných udalostí podľa ich priority do rozdielnych logovacích súborov hlavne kvôli prehľadnosti.

Veľmi dôležitá bola konfigurácia programu Postfix, ktorý slúži na prenos elektronickej pošty. Jeho konfigurácia bola kľúčová pre správne fungovanie ďalších súčastí servera, ktoré boli použité v tejto práci. Pri konfigurácii autentifikácie protokolu SMTP bol využitý moderný systém knižníc Cyrus-SASL. Antivírusová a antispamová kontrola všetkých e-mailov bola zabezpečená spoluprácou programov ClamAV, MailScanner a SpamAssassin. Ich konfigurácia bola úspešná a overená priamo v práci. Server bol nakonfigurovaný na prijímanie e-mailov zo slovenskej domény viphome.sk. Prijem e-mailov bol vyriešený pomocou doménového koša, ktorý bol vytvorený v administračnom rozhraní domény. Program Fetchmail bol nakonfigurovaný na pravidelné prenášanie všetkých správ pre túto doménu na server vytvorený v tejto práci. Jeho spúšťanie bolo zabezpečené pomocou programu cron. Všetky e-mailové súčasti spolupracujú s webmailom SquirrelMail. Jeho rozhranie bolo doplnené o dva veľmi používané doplnky - *autocomplete* a *retrieve user data*.

Server bol zabezpečený pomocou programu Netfilter. Na jednotlivých rozhraniach serveru boli povolené len určité TCP a UDP služby. Skript, ktorý sa stará o spúšťanie a zastavovanie jednotlivých príkazov iptables bol naprogramovaný v príkazovom interprete BASH a spúšťaný v úrovniach behu 2, 3 a 5. V závere bol nainštalovaný obľúbený grafický administračný nástroj Webmin.

Voľba systému Red Hat sa javila ako správna. Tento systém je veľmi spoľahlivý a dokumentácia k nemu veľmi dobrá. V priebehu testovacej prevádzky serveru nedošlo k žiadnym závažným výpadkom služieb a nebolo potrebné robiť v konfigurácii žiadne úpravy. Vzhľadom na obmedzený rozsah práce neboli konfigurované rôzne iné využiteľné časti, ktoré by ešte mohli zvýšiť bezpečnosť servera a zlepšiť jeho funkcionality ako sú TLS (Transport Layer Security), vylepšený webmail RoundCube alebo nástroj na grafickú správu programu Postfix - PostfixAdmin.

CONCLUSION

The complete e-mail server creating, including Red Hat system installation, SMTP, POP and IMAP server configuration and firewall script creation was described in this thesis. Red Hat system installation was simply and intuitive, the hardest part was creation of RAID partitions and LVM volumes. During the installation has not been used whole the space on the server hard drives, mainly because of later option to expand an existing partitions.

After the installation, at first a network connection was set up. It is very undesirable to run variety of well-known unnecessary services on the server and therefore these services were switched off. The system logging service syslog was configured for separate e-mail and authentication logging by the priority of the events to different logging files mainly to maintain the clarity.

The Postfix configuration was very important, which is used to mail transfer. This configuration was the key for right run of the next parts of the server that was used in this thesis. In the SMTP authentication configuration was used the modern Cyrus-SASL library. The antivirus and antispam control all of the e-mails was ensured by the cooperation of the ClamAV, MailScanner and SpamAssassin. Their configuration was successful and was verified directly in this work. Server was configured to receive e-mails from Slovak domain vipmohe.sk. E-mail receiving was ensured by the domain trash, which was created in domain administration interface. Fetchmail has been configured for periodically transferring all of the messages for this domain to server created in this work. His run was ensured by the cron daemon. All the e-mail components cooperate with webmail SquirrelMail. To its interface was added two very useful plugins – *autocomplete* and *retrieve user data*.

Server was secured with Netfilter. Only some TCP and UDP services were allowed on networking interfaces. The script, which takes care in starting and stopping the iptables instructions, has been programmed in BASH command interpreter and started up in runlevels 2, 3 and 5. In the end there were installed the favourite graphical administration tool Webmin.

The choice of Red Hat system seemed to be right. This system is very reliable and documentation is very good. There were none serious service interruptions and there was

no need to do configuration corrections during the server testing. Given the limited scope of this thesis, there were no other useful parts configured, which could improve the server security and improve its functionality, such as TLS (Transport Layer Security), RoundCube webmail or Postfix graphical management – PostfixAdmin.

ZOZNAM POUŽITEJ LITERATURY

- [1] A Brief History of Red Hat Linux. *Techotopia* [online]. 2010 [cit. 2013-03-12]. Dostupné z: http://www.techotopia.com/index.php/A_Brief_History_of_Red_Hat_Linux
- [2] DENT, Kyle D. *Postfix: kompletní průvodce*. 1. vyd. Praha: Grada, 2005, xiv, 237 s. ISBN 80-247-1029-3.
- [3] DOSEDĚL, Tomáš. *Počítačová bezpečnost a ochrana dat*. 1. vyd. Brno: Computer Press, 2004, 190 s. ISBN 80-251-0106-1.
- [4] Fichier:Red hat logo.png. *Wikipedia* [online]. 2005 [cit. 2013-03-12]. Dostupné z: http://fr.wikipedia.org/wiki/Fichier:Red_hat_logo.png#file
- [5] File Systems. *ArchLinux* [online]. 2013 [cit. 2013-03-12]. Dostupné z: https://wiki.archlinux.org/index.php/File_Systems
- [6] FLICKENGER, Rob. *Linux server na maximum: 100 tipů a řešení pro náročné*. Brno: CP Books, 2005, 229 s. ISBN 80-251-0586-5.
- [7] Historie OS UNIX. *Root.cz* [online]. 2001 [cit. 2013-03-12]. Dostupné z: <http://www.root.cz/clanky/historie-os-unix/>
- [8] Internet Message Format. *Network Working Group* [online]. 2001 [cit. 2013-03-12]. Dostupné z: <http://www.ietf.org/rfc/rfc2822.txt>
- [9] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.
- [10] KYSELA, Martin. *Přecházíme na Linux*. 1. vyd. Brno: Computer Press, 2003, 191 s. ISBN 80-722-6844-9.
- [11] *Linux: dokumentační projekt*. 4., aktualiz. vyd. Překlad Lubomír Ptáček. Brno: Computer Press, 2007, 1334 s. ISBN 978-80-251-1525-1.
- [12] Logovanie. ORAVEC, Miroslav. *Infoware* [online]. 2010 [cit. 2013-03-12]. Dostupné z: <http://www.itnews.sk/tituly/infoware/2010-06-19/c140436-logovanie>

- [13] SMITH, Roderick W. *Linux ve světě Windows: průvodce administrátora heterogenních sítí*. 1. vyd. Praha: Grada, 2006, xiii, 443 s. ISBN 80-247-1470-1.
- [14] Správa linuxového serveru: LVM a diskové šifrování. DOČEKAL, Michal. *Linux EXPRES* [online]. 2009 [cit. 2013-03-14]. Dostupné z: <http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-lvm-a-diskove-sifrovani>
- [15] Správa linuxového serveru: RAID teoreticky. DOČEKAL, Michal. *Linux EXPRES* [online]. 2009 [cit. 2013-03-12]. Dostupné z: <http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-raid-teoreticky>
- [16] Údržba logov - rotácia. *Infoware* [online]. 2010 [cit. 2013-03-13]. Dostupné z: <http://www.itnews.sk/tituly/infoware/2010-11-05/c140431-udrzba-logov-rotacia>
- [17] Činnosť firewallu. ORAVEC, Miroslav. *Infoware* [online]. 2008 [cit. 2013-03-14]. Dostupné z: <http://www.itnews.sk/tituly/infoware/2008-09-19/c140494-cinnost-firewallu>
- [18] Správa linuxového serveru: Linuxový firewall, základy iptables. DOČEKAL, Michal. *Linux EXPRES* [online]. 2010 [cit. 2013-03-15]. Dostupné z: <http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-linuxovy-firewall-zaklady-iptables>
- [19] Nástroj iptables. ORAVEC, Miroslav. *Infoware* [online]. 2008 [cit. 2013-03-15]. Dostupné z: <http://www.itnews.sk/tituly/infoware/2008-10-19/c140490-nastroj-iptables>
- [20] HILDEBRANDT, Ralf a Patrick KOETTER. *Postfix: provozujeme poštovní server v Linuxu*. 1. vyd. Brno: Computer Press, 2006, 431 s. ISBN 80-251-1020-6.

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

ACC	Antiques and Collectables of Connecticut
ARPANET	Advanced Research Projects Agency Network
ASCII	American Standard Code for Information Interchange
Cent OS	Community Enterprise Operating System
CRLF	Carriage Return Line Feed
DNS	Domain Name System
DSN	Delivery Status Notification
EPEL	Extra Packages for Enterprise Linux
ESMTP	Extended Simple Mail Transfer Protocol
EXT2	Second Extended File system
EXT3	Third Extended File system
EXT4	Fourth Extended File system
FSSTND	File System Standard
FTP	File Transfer Protocol
GNU GPL	GNU General Public License
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
LAN	Local Area Network
LHS	Left Hand Side
LVM	Logical Volume Manager
MDA	Mail Delivery Agent
MIME	Multipurpose Internet Mail Extension
MIT	Massachusetts Institute of Technology

MS-DOS	Microsoft Disc Operating System
MTA	Mail Transfer Agent
MUA	Mail User Agent
Multics	Multiplexed Information and Computing Service
NAT	Network Address Translation
NFS	Network File System
PAT	Port Address Translation
PHP	Hypertext Preprocessor
POP	Post Office Protocol
POSIX	Portable Operating System Interface
RAID	Redundant Array of Independent Disks
RFC	Request For Comments
RHEL	Red Hat Enterprise Linux
RHS	Right Hand Side
RPC	Remote Procedure Call
SASL	Simple Authentication and Security Layer
SMTP	Simple Mail Transfer Protocol
SENDMSG	Send Message
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOS	Type of Service
TTL	Time to Live
UDP	User Datagram Protocol
UIDL	Unique Mail Identifier

VERB	Verbose
WAN	Wide Area Network
XFS	Extended File System

ZOZNAM OBRÁZKOV

Obr. 1 – Logo distribúcie Red Hat [4].	16
Obr. 2 – Princíp činnosti LVM [14].	22
Obr. 3 – Cesta packetu v programe Netfilter.	28
Obr. 4 – Moderný e-mail.....	30
Obr. 5 – Princíp autentifikácie SMTP.	36
Obr. 6 – Server POP/IMAP.	38
Obr. 7 – Riešenie e-mailového servera.	41
Obr. 8 – Vytvorenie RAID partície.....	43
Obr. 9 – Vytvorenie LVM fyzického zväzku.....	44
Obr. 10 – Vytváranie jednotlivých logických zväzkov LVM.	44
Obr. 11 – Konfigurácia RAID, LVM, rozdelenie partícií.....	45
Obr. 12 – Detekcia vírusu a spamu v e-mailových správach.....	65
Obr. 13 – Konfiguračné rozhranie programu SquirrelMail.	71
Obr. 14 – Konfigurácia domény, SMTP a IMAP pre webmail.	72
Obr. 15 – Prihlasovací dialóg programu SquirrelMail.....	72
Obr. 16 – Prihlasovací dialóg programu Webmin.	76
Obr. 17 - Užívateľské konfiguračné rozhranie programu Webmin.	77

ZOZNAM TABULIEK

Tab. 1 – Typy súborov v Linuxe [11].	17
Tab. 2 – Odporúčaná konfigurácia diskových oddielov servera.	20
Tab. 3 – Úrovně behu systému Linux [11].	23
Tab. 4 – Příklady hodnôt parametrov „facility“ a „priority“	25
Tab. 5 – Tabuľky a reťaze programu Netfilter.....	27
Tab. 6 – Triedy stavových kódov protokolu SMTP.....	32
Tab. 7 – Konfigurácia sieťových kariet servera.	51

ZOZNAM PRÍLOH

P I Štruktúra zložiek disku uložená na CD

PRÍLOHA P I: ŠTRUKTÚRA ZLOŽIEK DISKU ULOŽENÁ NA CD

\Config\etc\	- obecné konfiguračné súbory z práce
\Config\etc\Dovecot\	- hlavný konfiguračný súbor servera Dovecot
\Config\etc\Dovecot\conf.d\	- konfiguračné súbory servera Dovecot
\Config\etc\httpd\conf\	- konfiguračný súbor servera Apache
\Config\etc\logrotate.d\	- konfiguračný súbor služby rotácie logov
\Config\etc\MailScanner	- konfiguračné súbory programu MailScanner
\Config\etc\postfix\	- konfiguračný súbor programu Postfix
\Config\etc\sasl2\	- konfiguračný súbor CyrusSASL
\Fetchmail	- konfiguračný súbor programu Fetchmail
\Firewall	- skript firewallu
\SquirrelMail	- doplnky programu SquirrelMail použité v práci
\TEXT	- táto práca v elektronickej podobe
\Webmin	- doplnky a vzhľady programu Webmin