UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

# Elektronická pošta, její zabezpečení a ochrana proti spamu

Michal Kašpar

Bakalářská práce
2012

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2011/2012**

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Michal Kašpar** |
| Osobní číslo: | **I09148** |
| Studijní program: | **B2646 Informační technologie** |
| Studijní obor: | **Informační technologie** |
| Název tématu: | **Elektronická pošta, její zabezpečení a ochrana proti spamu** |
| Zadávající katedra: | **Katedra informačních technologií** |

Zásady pro vypracování:

Cílem bakalářské práce je popis fungování elektronické pošty v síti. Budou představeny hrozby související s elektronickou poštou a bezpečnostní opatření pro zabezpečení e-mailu.
V úvodní teoretické části bude uveden popis fungování emailu (formáty zpráv dle IETF standardů), jak se zprávy v e-mailu přenášejí, zpřístupňují, uchovávají a související hrozby. Teoretická část bude obsahovat charakteristiku spamu a možnosti ochrany.
V praktické části bude instalován emailový server na VPS (Linux) včetně nastavení emailového klienta. Práce bude obsahovat ukázku podvržení odesílatele pomocí jednoduchého skriptu v PHP a jednoduché zachycení uživatelského hesla pomocí programu WireShark při použití nezabezpečené verze online emailového klienta.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**SOSINSKY, Barrie. Mistrovství - počítačové sítě. Praha : Computer Press, 2010. 840 s. ISBN 978-80-251-3363-7.**
**SCHRODER, Carla. Linux. Praha : Computer Press, 2009. 608 s. ISBN 978-80-251-2407-9.**
**KOCMAN, Rostislav; LOHNISKÝ, Jakub. Jak se bránit virům, spamu a spyware. Praha : Computer Press, 2005. 152 s. ISBN 80-251-0793-0.**
**R. STANEK, William. Microsoft Exchange Server 2010. Praha : ComputerPress, 2010. 696 s. ISBN 978-80-251-3342-2.**

Vedoucí bakalářské práce: **Ing. Soňa Neradová**
Katedra softwarových technologií

Datum zadání bakalářské práce: **16. prosince 2011**
Termín odevzdání bakalářské práce: **11. května 2012**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 30. března 2012

**Prohlášení autora**

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 4. 5. 2012 Michal Kašpar

**Poděkování**

Moje poděkování patří paní Ing. Soně Neradové za nápomoc a rady při vypracovávání této práce.

Dále děkuji rovněž panu Josefu Grillovi zastupujícím společnost WEDOS, a. s. za bezplatné poskytnutí virtuálního serveru (VPS) pro účely vypracování mé práce v reálných podmínkách.

V neposlední řadě bych rád poděkoval následujícím společnostem za poskytnutí spolupráce při vypracování průzkumu rozšíření spamu v roce 2012.

K nim patří:

- EBOLA Czech s. r. o.
- Economia, a. s.
- HostingSolutions s. r. o.
- Seznam.cz a. s.
- savana.cz s. r. o.
- WEBYA hosting, s. r. o.

**Anotace**

Cílem bakalářské práce je popis fungování elektronické pošty v síti. Budou představeny hrozby související s elektronickou poštou a bezpečnostní opatření pro zabezpečení e-mailu.

V úvodní teoretické části bude uveden popis fungování e-mailu (formáty zpráv dle IETF standardu), jak se zprávy v e-mailu přenášejí, zpřístupňují, uchovávají a související hrozby. Teoretická část bude obsahovat charakteristiku spamu a možnosti ochrany.

V praktické části bude instalován e-mailový server na VPS (Linux) včetně nastavení e-mailového klienta. Práce bude obsahovat ukázku odeslání zprávy s falešnou adresou odesílatele. Tato ukázka bude realizována pomocí skriptu v PHP. Dále bude názorně ukázáno zachycení uživatelského hesla pomocí programu Wireshark při použití nezabezpečené verze e-mailového klienta.

**Klíčová slova**

e-mail, elektronická pošta, spam, zabezpečení pošty

**Title**

Electronic mail, security and spam protection

**Annotation**

The purpose of this thesis is to describe functioning of electronic mail in computer networks. Then will be introduced threats related to e-mail and precautions to secure them.

In theoretical part will be described principles of e-mailing (message formats according to IETF standards), how messages are transfered, accessed, stored and threats related with e-mailing. This part will also include characteristics of spam and protection against it.

Practical part will contain installation of VPS (Linux) e-mail server and configuration of e-mail client. There will also be an illustration of sending e-mail message with fake sender address. This will be realized with plain PHP script. Then will be shown a caption of user password with Wireshark application while using a non-secured version of e-mail client.

**Keywords**

e-mail, electronic mail, spam, security

# Obsah

## Seznam zkratek

| | |
|---|---|
| ASCII | American Standard Code for Information Interchange |
| CSS | Cascading Style Sheets |
| DNS | Domain Name System |
| DoS | Denial of Service |
| E-MAIL | Electronic Mail |
| ESMTP | Extended Simple Mail Transfer Protocol |
| HTML | HyperText Markup Language |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| IMAP | Internet Message Access Protocol |
| IP | Internet Protocol |
| MDA | Mail Delivery Agent |
| MIME | Multipurpose Internet Mail Extensions |
| MTA | Mail Transfer Agent |
| MUA | Mail User Agent |
| PHP | PHP: Hypertext Preprocessor |
| POP | Post Office Protocol |
| RFC | Request for Comments |
| SMTP | Simple Mail Transfer Protocol |
| SSL | Secure Sockets Layer |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| VPS | Virtual Private Server |

## Seznam obrázků



## Seznam tabulek

# Úvod

Elektronická pošta neboli způsob přenosu elektronických zpráv prostřednictvím počítačové sítě je v dnešní době velmi rozšířený a využívaný způsob komunikace. Jedná se o způsob komunikace, který je pro koncové uživatele velmi jednoduchý k použití, avšak pro zajištění funkčnosti je zapotřebí jistého technologického zázemí a různých protokolů pro přenos dat. Tato práce se věnuje problematice elektronické pošty, jejího zabezpečení a hrozeb souvisejících s jejím užíváním.

V úvodu práce je popsáno, jakým způsobem je zajištěn samotný přenos e-mailu mezi koncovými uživateli. Dále je detailně rozebrána struktura e-mailové zprávy a možnosti při jejím vytváření včetně volitelných parametrů v záhlaví zprávy. První část práce se věnuje bližšímu popisu protokolů, které jsou pro přenos zprávy využívány. Jde o protokoly pro zpřístupnění zprávy koncovému uživateli, tak i o protokol zajišťující samotné odesílání.

Druhá část práce se věnuje zajištění bezpečnosti elektronické pošty. Na začátku této části jsou blíže popsány možnosti zabezpečeného přenosu zpráv nejen mezi servery, ale i mezi serverem a koncovým uživatelem. Praktická ukázka poukazuje na nebezpečí nezabezpečeného přístupu k e-mailové schránce.

Velkým problémem jsou e-mailové zprávy, které po uživatelích různými cestami požadují přihlašovací jména a hesla do různých systémů (například do internetového bankovnictví). Tyto e-maily vypadají často velmi věrohodně a na příkladu je názorně ukázáno, jak jednoduché je e-mailovou zprávu zfalšovat za pomocí skriptovacího jazyka PHP. Součástí této kapitoly je možnost využívání digitálních podpisů, které právě problém možného podvržení zprávy mají řešit.

Ve třetí části práce je blíže nastíněn problém, kterým je nevyžádaná pošta. Tato problematika je popsána od doby vzniku až po současnost. Na základě provedeného průzkumu jsou předložena aktuální data o průměrných počtech nevyžádané pošty, která proudí servery známých českých poskytovatelů. Taktéž jsou uvedeny různé metody, jak s nevyžádanou poštou, která uživatele natolik obtěžuje, vlastně bojovat.

Čtvrtá část obsahuje kompletní postup pro vytvoření vlastního e-mailového serveru na operačním systému Linux s veškerou potřebnou funkčností. S ohledem na předešlé části práce tedy nebyl nainstalován pouze základní e-mailový server, ale byly zprovozněny i další softwarové části potřebné pro zabezpečení komunikace. Součástí instalace je nasazení antispamového řešení. Výsledkem práce je vytvoření základního funkčního a zabezpečeného systému, který je možné využívat koncovým uživatelem.

# 1 Elektronická pošta

E-mail neboli elektronická pošta je způsob přenosu, odesílání a přijímání zpráv prostřednictvím počítačové sítě. Každý, kdo chce elektronickou poštu využívat, si vytvoří e-mailovou schránku, do které je mu tato pošta s pomocí e-mailové adresy zasílána.

## 1.1 Struktura zprávy

E-mailovou zprávu lze rozdělit na dvě hlavní oblasti. První je oblast takzvaných hlaviček neboli záhlaví zprávy. Je to oblast od začátku zprávy do prvního prázdného řádku. Vyskytují se zde jednotlivé záznamy (takzvané jednotlivé hlavičky). Mezi základní hlavičky patří například informace o předmětu zprávy, odesílateli a příjemci. Každá hlavička má jasně danou strukturu a nachází se na novém řádku. Prvním parametrem je klíčové slovo a za ním „:“ (například „FROM:“) [1], [2].

## 1.2 Oblast záhlaví

Záhlaví e-mailové zprávy obsahuje několik hlaviček, které slouží k identifikaci odesílatele a nastavení pravidel, na kterou e-mailovou adresu má přijít odpověď na zprávu.

- FROM - e-mailová adresa odesílatele e-mailu.
- SENDER - informace o skutečném odesílateli e-mailu, pokud se liší od adresy uvedené v hlavičce FROM.
- REPLY-TO - adresa pro zaslání odpovědi. Tato hlavička je využívána, pokud má odpověď přijít na jinou adresu než na tu, která je v hlavičce FROM [1], [2].

Tyto 3 hlavičky mohou vypadat jako podobné, ale drobně se liší. Nastane-li chyba při transportu e-mailu, tak se tato chyba zasílá na adresu uloženou v hlavičce SENDER. Na tuto hlavičku se nezasílá odpověď na e-mailovou zprávu (je možné na ni zaslat odpověď pouze explicitně, je-li nutné kontaktovat zodpovědnou osobu). Je-li nastavená hlavička REPLY-TO, pak má přednost před polem FROM, jinak se odpověď zasílána adresu uvedenou v hlavičce FROM [1], [2].

Záhlaví zprávy je však daleko pestřejší, a tak nabízí dále například tyto hlavičky.

- SUBJECT - předmět e-mailu.
- TO - e-mailová adresa příjemce e-mailu.
- CC - e-mailové adresy pro zaslání kopií zprávy. Tyto adresy jsou ve zprávě viditelné.
- BCC - adresa, na které se mají zaslat skryté kopie e-mailu. Využívá se, pokud chceme zaslat skrytou kopii na nějakou adresu, primární adresát tuto adresu ve zprávě nevidí.
- MESSAGE-ID – jedinečný identifikátor zprávy.
- RECEIVED - hlavička přidávaná při každém zpracovávání během transportu e-mailu.

- RETURN-PATH - zpáteční adresa k odesílateli e-mailu.
- DELIVERED-TO - informace komu je zpráva doručována.
- DATE - datum a čas odeslání e-mailu [1], [2].

Ze seznamu je patrné, že položek v oblasti hlaviček může být poměrně velké množství a to se jedná pouze o výčet těch nejzákladnějších. Velká část hlaviček není ve výsledku uživateli v jeho e-mailovém klientu vůbec zobrazena, část není povinná. Ve většině případů e-mail obsahuje tyto hlavičky, které by se daly označit za ty základní - FROM (odesílatel zprávy), TO (příjemce zprávy), SUBJECT (předmět zprávy) a DATE (datum odeslání zprávy, které se automaticky vyplní e-mailovým klientem v okamžiku odeslání) [1], [2].

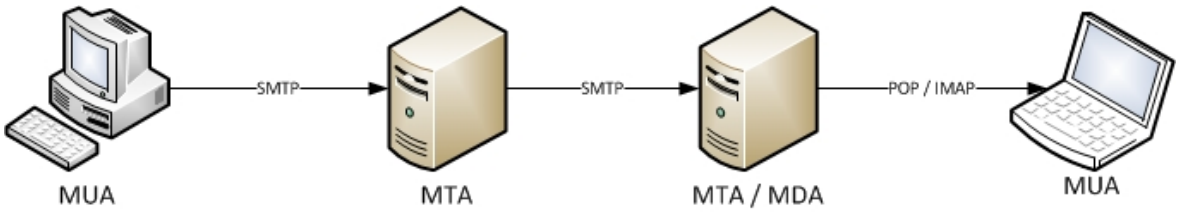
## 1.3 Účastníci přenosu

Na přenosu elektronické pošty mezi dvěma koncovými uživateli se podílí několik typů účastníků, kteří se dělí do následujících skupin.

**MUA (Mail User Agent)** - do této skupiny patří nástroje pro práci se schránkou, prostřednictvím nichž uživatel zadává své požadavky. Patří sem on-line e-mailoví klienti, ale i desktopové aplikace jako například Microsoft Outlook, Mozilla Thunderbird a další [3].

**MTA (Mail Transfer Agent)** - jedná se o server starající se o doručení zprávy na cílový server adresáta zprávy. Do skupiny MTA je možné zařadit různé e-mailové servery (Postfix, Sendmail, Exim, Qmail či Microsoft Exchange Server) [3].

**MDA (Mail Delivery Agent)** - neboli nástroj zodpovědný za lokální doručení. Pod pojmem lokální doručení je myšleno vlastní zařazení e-mailové zprávy do příslušné e-mailové schránky jejího adresáta. Skupina MDA je zavedená z toho důvodu, že nabízí efektivnější zařazování do jednotlivých e-mailových schránek. Do této kategorie patří například Maildrop či Procmail [3].




**Obrázek 1 - Účastníci přenosu e-mailu**

## 1.4 Průběh přenosu zprávy a vznik jednotlivých hlaviček

Pro lepší názornost přenosu elektronické pošty bude v následujícím příkladu simulováno odeslání e-mailové zprávy z adresy e-mail@michalkaspar.cz, která využívá e-mailové servery společnosti WEDOS, a. s. E-mailová zpráva bude odeslána na adresu

st28556@student.upce.cz, která je hostována u společnosti Google Inc. Bude uveden krok po kroku postup, jak vznikaly jednotlivé hlavičky u přenášené e-mailové zprávy.

V době psaní zprávy, tedy ještě před odesláním, byla oblast hlaviček e-mailu velmi jednoduchá a obsahovala pouze tyto následující záznamy. Hlavička FROM určuje adresu odesílatele, TO udává e-mailovou adresu příjemce, v hlavičce SUBJECT se nachází předmět zprávy a poslední hlavička DATE je vyplněna v okamžik odeslání přímo e-mailovým klientem, který jí přiřadí lokální uživatelský čas [4].

```
Date: Wed, 8 Feb 2012 14:58:43 +0100
From:e-mail@michalkaspar.cz
To: st28556@student.upce.cz
Subject: Predmet zpravy
```

Po příchodu na server společnosti WEDOS, a. s. je do oblasti záhlaví zprávy přidána hlavička RECEIVED, která udává informaci, odkud byla zpráva přijata, prostřednictvím kterého serveru, pro koho je zpráva určena a také je zde časový údaj přijetí e-mailové zprávy na server. Jelikož bylo pro spojení se SMTP serverem použito zabezpečené SSL spojení, tak se zde nachází i údaj o zabezpečeném spojení. Druhou hlavičkou, která přibyla je MESSAGE-ID, což je jedinečný identifikátor zprávy přidělený serverem sloužící k její identifikaci [4].

```
Received: from MichalNB ([77.48.237.92])
        by mail1.wedos.net (WEDOS Mail Server mail1) with ASMTP (SSL) id
SRO00144
        for <st28556@student.upce.cz>; Wed, 08 Feb 2012 14:58:44 +0100
Message-ID: <001801cce669$c54d6500$4fe82f00$@michalkaspar.cz>
```

V dalším kroku je vidět, jak přibývaly další hlavičky e-mailové zprávy. Jedná se o okamžik, kdy e-mailová zpráva byla přeposlána ze serverů společnosti WEDOS, a. s. na servery společnosti Google Inc. [4].

```
Delivered-To: st28556@student.upce.cz
Received: by 10.216.182.144 with SMTP id o16cs32092wem;
        Wed, 8 Feb 2012 05:58:45 -0800 (PST)
Received: by 10.14.48.65 with SMTP id u41mr6445056eeb.39.1328709525602;
        Wed, 08 Feb 2012 05:58:45 -0800 (PST)
Return-Path: <e-mail@michalkaspar.cz>
Received: from mail1.wedos.net (mail1.wedos.net. [46.28.105.6])
        by mx.google.com with ESMTPS id
z48si1065417eeb.196.2012.02.08.05.58.44
        (version=TLSv1/SSLv3 cipher=OTHER);
        Wed, 08 Feb 2012 05:58:45 -0800 (PST)
```

Aby bylo jasné, co se v tomto bodě stalo, je nutné číst tyto hlavičky zespodu nahoru. Přesun zprávy na servery společnosti Google Inc. ukazuje poslední z hlaviček RECIEVED. Hlavička poskytuje informaci, že server mail1.wedos.net s konkrétní IP adresou 46.28.105.6 zaslal tuto zprávu na servery společnosti Google Inc. Je vidět, že pro přenos zprávy bylo použito zabezpečené spojení. V tomto případě se jednalo o spojení TLS. Položka RETURN-PATH uchovává adresu původního odesílatele e-mailové zprávy. Následně byly přidány dvě hlavičky RECIEVED, které informují o konečném přijetí

zprávy na servery společnosti Google Inc. a přesuny v rámci těchto serverů. Vždy je opět vidět IP adresa, přes kterou zpráva přešla, a čas, kdy se tak stalo. Poslední přidanou hlavičkou, v tomto případě první záznam v bloku kódu nesoucí označení DELIVERED-TO udává pro koho je zpráva určena, tudíž do které e-mailové schránky má být umístěna [4].

Sestavování hlaviček e-mailové zprávy je celkem jednoduchý postup, kdy se zaznamenává každý přesun zprávy a hlavičky se vkládají vždy na začátek oblasti hlaviček. Sestavíme-li z těchto kroků kompletní e-mailovou zprávu, tak ta po přijetí do konečné e-mailové schránky vypadala následovně:

```
Delivered-To: st28556@student.upce.cz
Received: by 10.216.182.144 with SMTP id o16cs32092wem;
        Wed, 8 Feb 2012 05:58:45 -0800 (PST)
Received: by 10.14.48.65 with SMTP id u41mr6445056eeb.39.1328709525602;
        Wed, 08 Feb 2012 05:58:45 -0800 (PST)
Return-Path: <e-mail@michalkaspar.cz>
Received: from mail1.wedos.net (mail1.wedos.net. [46.28.105.6])
        by mx.google.com with ESMTPS id
z48si1065417eeb.196.2012.02.08.05.58.44
        (version=TLSv1/SSLv3 cipher=OTHER);
        Wed, 08 Feb 2012 05:58:45 -0800 (PST)
Received: from MichalNB ([77.48.237.92])
        by mail1.wedos.net (WEDOS Mail Server mail1) with ASMTP (SSL) id
SRO00144
        for <st28556@student.upce.cz>; Wed, 08 Feb 2012 14:58:44 +0100
Message-ID: <001801cce669$c54d6500$4fe82f00$@michalkaspar.cz>
Date: Wed, 8 Feb 2012 14:58:43 +0100
From:e-mail@michalkaspar.cz
To: st28556@student.upce.cz
Subject: Predmet zpravy

Toto je text zpravy.
```

V kompletním kódu e-mailové zprávy je vidět, že oblast záhlaví je opravdu oddělena od těla zprávy pouze prázdným řádkem. Tělo testovací zprávy neslo pouze krátký testovací text. Zpráva byla odeslána jako prostý text [4].

## 1.5 Multipurpose Internet Mail Extensions

MIME (Multipurpose Internet Mail Extensions) je rozšíření klasického formátu elektronické pošty se snahou řešit nedostatky a přinést potřebná rozšíření. Formát MIME je definován v RFC 1521 a RFC 1522 [5].

Základním problémem klasického formátu elektronické pošty je to, že byl navrhován pro přenos zpráv v jazycích, kterým stačí pouze znaky ASCII. Tato situace lze určitými metodami řešit, ale bylo nutné zavést celosvětový standard [5].

MIME formát rozšiřuje možnosti klasického e-mailu o podporu textu psaného v jiné znakové sadě než US-ASCII, zavádí možnost vícedílných zpráv, přidává možnost práce s přílohami (obrázky, dokumenty, atd.) a v neposlední řadě zavádí možnost uvést i informaci v samotné hlavičce v jiné znakové sadě než ASCII [5].

V dnešní době se tedy používají převážně e-mailové zprávy v tomto formátu. Ukázka zdrojového kódu takové zprávy je následující. Pro zjednodušení a snahy ukázat pouze rozdíly oproti klasické e-mailové zprávě, je tato ukázková zpráva odeslána na vlastní e-mailovou adresu. Nebude zde tedy vidět žádný průchod servery jako v předchozím příkladu.

```
MIME-Version: 1.0
Received: by 10.216.182.144 with HTTP; Wed, 8 Feb 2012 07:09:40 -0800
(PST)
Date: Wed, 8 Feb 2012 16:09:40 +0100
Delivered-To: st28556@student.upce.cz
Message-ID:
<CA+6rhFvTgoBqycuLiDCe55BUXYWUpD6+O3rs+twxr=LQYtdbUw@mail.gmail.com>
Subject: =?ISO-8859-2?Q?Pokusn=E1_zpr=E1va_=E8=EDslo_2?=
From: =?ISO-8859-2?Q?Michal_KA=A9PAR?= <st28556@student.upce.cz>
To: =?ISO-8859-2?Q?Michal_KA=A9PAR?= <st28556@student.upce.cz>
Content-Type: multipart/alternative;
boundary=001636c5b9e0e79e7404b875489d

--001636c5b9e0e79e7404b875489d
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable

Toto je druh=E1 pokusn=E1 zpr=E1va, tentokr=E1t odesl=E1na jako MIME.

--001636c5b9e0e79e7404b875489d
Content-Type: text/html; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable

Toto je druh=E1 pokusn=E1 zpr=E1va, tentokr=E1t odesl=E1na jako MIME.

--001636c5b9e0e79e7404b875489d--
```

Z kódu je patrné, že dochází k rozšíření možností standardního formátu elektronické pošty. Na prvním řádku oblasti hlaviček je uvedena verze MIME - MIME-Version: 1.0. Dále je zde uveden typ zprávy - Content-Type:text/html, což značí, že se jedná o textovou zprávu, které je možné formátovat značkami HTML jazyka. Je tak možné zprávu formátovat za pomocí tagů jazyka HTML. Také se zde nachází informace o použité znakové sadě zprávy - charset=ISO-8859-1.

Zajímavostí je hlavička, jejíž data obsahují diakritiku. V ukázkové zprávě například předmět e-mailu.

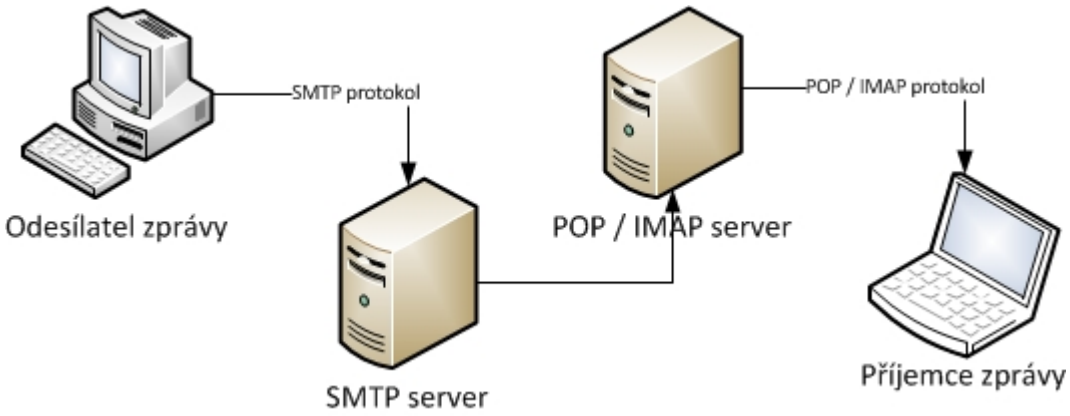
```
Subject: =?ISO-8859-2?Q?Pokusn=E1_zpr=E1va_=E8=EDslo_2?=
```

Pomocí tohoto zápisu je zajištěna možnost použití diakritiky i v samotných hlavičkách zprávy. Používá se zde takzvaná Encoded-word syntaxe. V této syntaxi je uveden originální text převedený do ASCII, cílová znaková sada a dekódovací metoda. Díky těmto parametrům se text zobrazí správně a je tak možné využívat znaky mimo ASCII.

# 2 Protokoly elektronické pošty

Elektronická pošta je závislá nejen na doručování zpráv mezi servery, na kterých je schránka odesílatele a servery, na kterých je schránka příjemce zprávy. Taktéž je závislá na možnosti přístupu do vlastní e-mailové schránky z koncového zařízení uživatele (PC, notebook, smartphone a další). Pro zajištění těchto funkcionalit je zapotřebí příslušných síťových protokolů.

Tato kapitola tedy bude zaměřena na protokoly sloužící k přístupu k e-mailové schránce (POP, IMAP) a protokol sloužící pro odesílání e-mailové zprávy (SMTP). Využití jednotlivých protokolů je znázorněno na následujícím obrázku [6].




**Obrázek 2 - Protokoly použité k přenosu e-mailu**

## 2.1 Post Office Protocol

Post Office Protocol (POP) je prvním z protokolů využívaných při práci s elektronickou poštou. Je určen pro příjem pošty ze vzdáleného serveru, na kterém je umístěna poštovní schránka uživatele do e-mailového klienta. Jedná se o aplikační protokol využívající TCP/IP spojení na portu 110[1] [1], [6].

Princip funkce tohoto protokolu je takový, že za pomocí série textových dotazů a odpovědí je zjištěno, zda jsou ve schránce nějaké e-mailové zprávy. Tyto zprávy jsou staženy a zároveň originály ze vzdáleného serveru smazány (ve 3. verzi, je již možnost e-mailové zprávy ze serveru nemazat). Toto typické chování protokolu POP je však problém při potřebě synchronizace zpráv mezi více zařízeními [1], [6].

V aktuální době je využíván protokol POP ve verzi 3, který je standardizován v RFC 1939. Tato verze protokolu POP se od předchozích verzí poměrně liší. Při jejím používání je v e-mailovém klientu možnost mnoha nastavení. Příkladem je možnost zanechávat originály e-mailových zpráv na vzdáleném serveru bez smazání. Dále byla doplněna možnost stahovat pouze hlavičky e-mailových zpráv. Tato vlastnost není v některých e-mailových klientech podporována [1], [7].

[1] Port 110 je standartním nastavením protokolu POP v případě nezabezpečené verze.

### 2.1.1 Spojení pomocí protokolu POP

Princip funkce POP3 spojení je následující. Klient vytvoří TCP spojení na portu, na kterém běží POP3 server. Po vytvoření spojení a obdržení uvítací zprávy od serveru může klient zasílat další příkazy a pracovat tak s e-mailovou schránkou [1], [7].

Příkazy u protokolu POP3 mají 3 až 4 písmena a začínají na novém řádku. Za příkazem samotným následují argumenty, které jsou oddělené mezerami. Příkazy i jejich argumenty jsou složené z ASCII znaků a jsou ukončeny vždy za pomocí CRLF [1], [7].

Server klientovi vždy zasílá odpovědi ve tvaru identifikátoru stavu a případných dodatečných informací. Identifikátory stavu jsou 2 a to +OK nebo -ERR. V případě delší odpovědi server odešle více řádků následovaných řádkem CRLF [1], [7].

Spojení za pomocí protokolu POP3 prochází několika stavy.

**Autorizační fáze** - po zahájení TCP spojení je server ve stavu autorizace. Ačkoliv protokol jako takový přímo nepředepisuje žádnou formu autorizace, tak nejrozšířenější je metoda autorizace USES/PASS. Klient nejdříve v tomto stavu zašle přihlašovací jméno (příkaz USER), odpoví-li server kladně, tak následuje zaslání přihlašovacího hesla (příkaz PASS). Pokud kombinace obou údajů odpovídá údajům na serveru, tak je spojení autorizováno. Základním problémem bezpečnosti je, že výchozí spojení není nijak zabezpečeno. Přihlašovací jméno i heslo jsou tak přenášeny v nezabezpečené podobně a tak není příliš velkým problémem tato data odposlechnout. Tento problém řeší šifrování spojení pomocí SSL / TLS [1], [7].

**Transakční fáze** - po úspěšné autorizaci následuje fáze transakční. Tato fáze je z pohledu práce s e-mailovou schránkou nejdůležitější, zde s ní klient totiž pracuje. Může zasílat například příkazy STAT, RETR, DELETE, LIST a další. Všechny změny jsou v této fázi prováděny virtuálně - k jejich skutečné realizaci dojde až ve fázi aktualizace [1], [7].

**Fáze aktualizace** - jakmile klient v transakční fázi zadá příkaz QUIT, tak server přechází do třetí a poslední fáze nazývané fáze aktualizace (v originálním znění fáze update). Zde server uvolní rezervované prostředky a dokončí práci se schránkou (např. smaže zprávy označené jako ke smazání, atd.) [1], [7].

### 2.1.2 Základní přehled příkazů protokolu POP

- USER uživatelské_jméno - slouží k zadání uživatelského jména.
- PASS uživatelské_heslo - slouží k zadání uživatelského hesla.
- STAT - příkaz pro zjištění aktuálního stavu komunikace. Tento příkaz vypíše informace o schránce - počet zpráv a jejich velikost (velikost je udávána v oktetech). Například pomocí odpovědi “+OK 2 300” server informuje, že ve schránce jsou 2 zprávy o celkové velikosti 300 oktetů.

- LIST [číslo] - tento příkaz s volitelným parametrem vypíše umožnuje vypsat počet zpráv na serveru a jejich velikost. Pomocí volitelného argumentu však můžeme zajistit vypsání velikosti konkrétní zprávy.
- RETR číslo - pomocí tohoto příkazu zajistíme přenos zprávy s příslušným číslem ze serveru ke klientovi. Argument obsahující číslo zprávy je zde povinný.
- DELE číslo - tento příkaz zajistí označení zprávy s příslušným číslem jako zprávu ke smazání. Argument obsahující číslo zprávy je zde povinný.
- RSET - umožnuje předčasně ukončit aktuální poštovní transakci (v tomto případě dojde například k odznačení zpráv určených ke smazání).
- NOOP - nemá žádnou funkci. Používá se pouze k otestování spojení se serverem. Server v tomto případě zasílá odpověď +OK.
- APOP identifikátor_schránky md5_řetězec - volitelný příkaz využívaný ve fázi autorizace uživatele na server. Prvním argumentem je identifikátor poštovní schránky, druhým je MD5 řetězec definovaný v dokumentaci RFC 1321. Oba argumenty jsou v tomto případě povinné.
- TOP zpráva počet_řádků - pro získání části zprávy. Argument zpráva slouží k její identifikaci a argument počet_řádků udává počet řádků z těla zprávy. Pomocí tohoto příkazu je tedy možné získat záhlaví zprávy a pouze několik řádků z těla zprávy.
- QUIT - slouží k ukončení spojení. V případě zadání ve fázi autorizace dojde k ukončení spojení, v případě zadání ve fázi transakční přechází server do fáze aktualizace [1], [7].

### 2.1.3 Ukázka spojení pomocí protokolu POP3

V této praktické ukázce spojení server (označen jako S) naslouchá na definovaném portu, dokud se nepřihlásí nějaký klient (označen jako K). Následuje přihlášení za pomocí příkazů USER a PASS. Dále je pomocí příkazů STAT a LIST vypsán obsah schránky.

Ve schránce se nachází pouze 1 e-mailové zpráva a tak je za pomocí příkazu RETR stažena ke klientovi. Následně je pomocí příkazu DELE označena příznakem pro smazání. Po zadání příkazu QUIT klientem je ukončena transakční fáze spojení.

```
S: +OK Dovecot ready.
K: USER uzivatel
S: +OK
K: PASS heslo
S: +OK Logged in.
K: STAT
S: +OK 1 1796
K: LIST
S: +OK 1 messages:
S: 1 1796
S: .
K: RETR 1
S: +OK 1796 octets
S: <Vypsán kompletní obsah e-mailové zprávy>
S: .
K: DELE 1
```

```
S: +OK Marked to be deleted.
K: QUIT
```

## 2.2 Internet Message Access Protocol

Internet Message Access Protocol (IMAP) je dalším z protokolů sloužících ke vzdálenému přístupu k elektronické poště. Využívá TCP/IP spojení na portu 143[2]. Protokol IMAP přináší daleko více možností pro práci se schránkou [1], [8].

Pracuje se zprávami přímo na serveru v režimu dlouhodobého připojení. Do e-mailového klienta se stahují pouze nezbytné informace, což znamená, že při zobrazení složky zpráv se tedy stáhnou pouze jejich záhlaví, až při zobrazení konkrétní zprávy se tato zpráva stáhne do e-mailového klienta [1], [8].

Tento protokol je navržen tak, aby se zprávami na vzdáleném serveru pracoval jako by byly uloženy na lokálním disku. E-mailový klient tak může zprávy přesouvat mezi schránkami, editovat, ukládat, vyhledávat a samozřejmě i mazat. Umí pracovat v on-line i off-line režimu (v tomto případě pouze s předem načtenými daty). Protokol IMAP je velmi vhodný pří práci v e-mailové schránce z více zařízení (možno i zároveň) [1], [8].

Díky velmi rozsáhlým možnostem je protokol IMAP daleko složitější a náročnější na server, což může občas způsobovat pomalejší komunikaci se serverem.

V současné době se používá protokol IMAP4rev1. Tato verze je definována v normě RFC 3501 a je zpětně kompatibilní s IMAP2, IMAP2bis i IMAP4. Oproti starším verzím má implementovánu podporu šifrovaného přihlašování. Je však možný i přenos nezabezpečeného hesla zejména z důvodů, že šifrovaný mechanismus musí být odsouhlasený klientem i serverem, což v případě Linuxových serverů a Windowsových klientů může být problémem. Tento problém je však stejně jako u protokolu POP3 možné obejít použitím šifrovaného spojení použitím SSL nebo TLS.

### 2.2.1 Hlavní změny oproti protokolu POP

**Více klientů současně** - protokol IMAP umožňuje oproti protokolu POP připojení více klientů najednou a synchronizaci změn provedených libovolným klientem [1], [8].

**Trvalé připojení** - protokol POP3 se připojuje k serveru pouze na nezbytně dlouhou dobu na to, aby stáhl zprávy do e-mailového klienta. Oproti tomu u protokolu IMAP je klient připojen k serveru trvale, dokud je aktivní například uživatelské rozhraní klienta [1], [8].

**Příznaky u zpráv a jejich synchronizace** - zvláště pokud má uživatel ve schránce velké množství zpráv a rozhodne se používat u jednotlivých zpráv takzvané příznaky (například u e-mailů, které je nutné zpracovat apod.), tak je protokol IMAP velmi vhodný. Příznaky se tak totiž ukládají na serveru a synchronizují se mezi jednotlivými klienty [1], [8].

[2] Port 143 je standartním nastavením protokolu IMAP v případě nezabezpečené verze.

**Vyhledávání přímo na serveru** - protokol IMAP pracuje se zprávami přímo na serveru a umožňuje tak přímo v e-mailové schránce na serveru i vyhledávat. U protokolu POP3 bylo nutné všechny zprávy nejprve stáhnout a poté v nich až lokálně vyhledávat pomocí funkcionality e-mailového klienta [1], [8].

**Způsob přístupu ke zprávám MIME** - protokol IMAP nabízí v případě zpráv MIME možnost stahování pouze její části [1], [8].

**Rozsáhlá možnost práce se zprávami** - IMAP díky své vlastnosti trvalého připojení a práci přímo s e-maily na serveru umožňuje jejich přesouvání mezi složkami a další možnosti úprav [1], [8].

### 2.2.2 Spojení pomocí protokolu IMAP

Ve spojení pomocí protokolu IMAP zahajuje klient komunikaci tak, že odesílá řádek začínající tagem (v protokolu IMAP představuje tag identifikátor příkazu), za nímž následuje klíčové slovo a další argumenty. Může nastat situace, že odeslaný řádek není kompletním příkazem, server v takovém případě odešle žádost o doplnění s prefixem +. Pokud je příkaz již kompletní, server přečte a zpracuje příkaz, na základě něhož vrátí požadovaná data. Ta mají v protokolu IMAP prefix * [1], [8].

Stejně jako při spojení za pomocí protokolu POP3 i při spojení za pomocí protokolu IMAP prochází toto spojení několika stavy.

**Non-authenticated** - jedná se o výchozí stav, ve kterém je klient v okamžiku, kdy ještě neproběhla autentizace. Klient se tedy musí nejprve autentizovat u serveru, aby mohl provádět další příkazy [1], [8].

**Authenticated** - po úspěšné autentizaci klient přechází do tohoto stavu, kde je nutné vybrat poštovní schránku, se kterou chce klient pracovat [1], [8].

**Selected** - v tomto stavu již klient byl úspěšně autentizován a proběhl výběr schránky, schránka je vybrána a je možné s ní dále pracovat a provádět s ní další operace [1], [8].

**Logout** - tento stav může být vyvolán příkazem zadaným klientem, ale i rozhodnutím serveru. Při přerušení spojení server automaticky přechází do tohoto stavu [1], [8].

### 2.2.3 Základní přehled příkazů protokolu IMAP

- LOGIN - slouží k přihlášení uživatele pomocí uživatelského jména a hesla.
- CREATE - vytvoří poštovní schránku (složku).
- DELETE - smaže poštovní schránku (složku).
- RENAME - přejmenuje poštovní schránku (složku).
- LIST - slouží k vypsání obsahu složky. Má dva parametry - prvním je cesta ke složce a druhým je jméno poštovní schránky.
- SELECT - příkaz k načtení /otevření poštovní schránky, jejíž jméno je dáno jako parametr příkazu.

- EXAMINE - pracuje jako SELECT s rozdílem, že otevře schránku pouze ke čtení.
- CLOSE - pomocí tohoto příkazu je zavřena poštovní schránka a spojení se vrací do stavu před příkazem SELECT.
- COPY - slouží pro kopírování z otevřené poštovní schránky do druhé, která je dána jako parametr.
- FETCH - určen pro stažení zprávy do e-mailového klienta, případně její části.
- STORE - příkaz sloužící k nastavení příznaku zprávy.
- SEARCH - slouží k vyhledávání v poštovní schránce. Vyhledává se podle parametru zadaného v příkaze. Při více parametrech se vyhledává na základě (AND, OR, NOT).
- LOGOUT - slouží k odhlášení ze schránky [1], [8].

### 2.2.4 Ukázka spojení pomocí protokolu IMAP4rev1

V této praktické ukázce dochází nejprve k přihlášení klienta (označen jako K) za pomocí příkazu LOGIN na server (označen jako S). Následně je za pomocí příkazu SELECT nastavena aktivní schránka (inbox - doručená pošta). Následuje stáhnutí hlavičky první zprávy za pomoci příkazu FETCH 1 BODY[HEADER]. Na závěr je příkazem STORE nastaven této zprávě příznak smazané zprávy a provedeno odhlášení za pomoci příkazu LOGOUT.

```
S: * OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE
STARTTLS A
UTH=PLAIN AUTH=LOGIN] Dovecot ready.
K: LOGIN uzivatel heslo
S: OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE
SORT SORT=DI
SPLAY THREAD=REFERENCES THREAD=REFS MULTIAPPEND UNSELECT IDLE CHILDREN
NAMESPACE
UIDPLUS LIST-EXTENDED I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT
SEARCHRES WIT
HIN CONTEXT=SEARCH LIST-STATUS] Logged in
K: a002 select inbox
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags
permitted.
* 2 EXISTS
* 0 RECENT
* OK [UNSEEN 2] First unseen.
* OK [UIDVALIDITY 1333199430] UIDs valid
* OK [UIDNEXT 31] Predicted next UID
* OK [HIGHESTMODSEQ 1] Highest
S: a002 OK [READ-WRITE] Select completed.
C: a003 fetch 1 body[header]
S: * 1 FETCH (BODY[HEADER] {1275}
S: <Vypsána hlavička e-mailové zprávy>
S:
S: )
S: a003 OK Fetch completed.
K: a004 store 1 +flags \deleted
S: * 1 FETCH (FLAGS (\Deleted \Seen))
a005 OK Store completed.
K: a005 logout
```

## 2.3 Simple Mail Transfer Protocol

Simple Mail Transfer Protocol (SMTP) funguje na síťovém protokolu TCP/IP na portu 25[3]. Oproti dvěma předchozím protokolům (POP, IMAP) se protokol SMTP používá pro odesílání zpráv. Jedná se o poměrně starý protokol, který je definován v normě RCF 821. Tuto normu následně rozšířila novější norma RFC 2821 [1], [9].

### 2.3.1 Spojení pomocí protokolu SMTP

SMTP spojení funguje tak, že na základě požadavku klienta vytvoří odesílající SMTP server spojení. Jelikož je SMTP protokolem typu požadavek odpověď jako předchozí protokoly, tak klient po vytvoření spojení zasílá jednotlivé požadavky, na které server odpovídá [1], [9].

Mezi základní příkazy používané v rámci této komunikace patří.

- HELO - příkaz, který se používá při navazování spojení. Za tímto příkazem uvede klient serveru svoji identifikaci.
- MAIL - je zadáván hned na začátku poštovní transakce a používá se v kombinaci s klíčovým slovem FROM, kde je uvedena adresa odesílatele.
- RCPT - slouží jako opak předchozího příkazu. Používá se s klíčovým slovem TO a uvádí se zde adresát zprávy. V případě více adresátů zprávy se příkaz zadává vícekrát.
- DATA - příkaz DATA identifikuje začátek sekce, kde zasíláme kompletní e-mailovou zprávu včetně hlaviček serveru SMTP.
- NOOP - příkaz stejně jako u předchozích protokolů nemá žádnou přesně definovanou funkci. Po jeho zadání následuje odpověď serveru. Může tak sloužit ke zjištění stavu spojení.
- RSET - umožňuje předčasné ukončení transakce se serverem bez jejího dokončení (odeslání e-mailu).
- QUIT - ukončí spojení se serverem. Zpráva je odeslána [1], [9].

### 2.3.2 Ukázka spojení pomocí protokolu SMTP

Tento příklad ukazuje způsob, jakým je odeslána zpráva od klienta (označen jako K), za pomoci serveru (označen jako S) prostřednictvím protokolu SMTP. Na začátku ukázky je úvodní příkaz HELO. Následně je sdělena adresa odesílatele příkazem MAIL FROM a adresa příjemce za pomocí příkazu RCPT TO. Pokračuje příkaz DATA, po kterém je vložen kód kompletní e-mailové zprávy. Nakonec příkaz QUIT ukončuje spojení.

```
S: 220 mail.testmail.cz ESMTP Postfix (Debian/GNU)
K: HELO ukazka
S: 250 mail.testmail.cz
K: MAIL FROM:<email@michalkaspar.cz>
S: 250 2.1.0 Ok
K: RCPT TO:<michal@testmail.cz>
```

[3] Port 25 je standartním nastavením protokolu SMTP v případě nezabezpečené verze.

```
S: 250 2.1.5 Ok
K: DATA
S: 354 End data with <CR><LF>.<CR><LF>
K: From: "Michal Kaspar" <email@michalkaspar.cz>
K: To: "Michal Kaspar" <michal@testmail.cz>
K: Date: Sat, 7 Apr 2012 18:00:00 +0000
K: Subject: Pokusny email
K:
K: Toto je obsah pokusneho emailu.
K: .
S: 250 2.0.0 Ok: queued as CC8193C0E2
K: QUIT
```

## 2.4 Extended Simple Mail Transfer Protocol

Extended Simple Mail Transfer Protocol (ESMTP) je rozšířením standartního protokolu SMTP. Poprvé byl definován v roce 1995 v dokumentu RCF 1869, následně se stal součástí dokumentu RFC 2821 [1], [9].

Tento protokol zavádí některé nové příkazy. Příkladem je příkaz EHLO. Tímto příkazem se zahajuje spojení ESMTP. Server na ně oproti klasickému SMTP odpoví i kompletní výčet povolených příkazů pro komunikaci s konkrétním serverem [1], [9].

Je zajištěna zpětná kompatibilita se standartním SMTP protokolem. Klient na začátku spojení zkusí zaslat příkaz EHLO. Odpoví-li server kladně s kódem 250, tak může pokračovat ve využívání ESMTP. Odpoví-li s chybou (typicky 500), tak klient odesílá starší příkaz HELO [1], [9].

Mezi další příkazy, které jsou zavedené v protokolu ESMTP patří například:

- SMTP-AUTH - umožní odeslat zprávy až poté, co se uživatel přihlásí.
- SIZE - server udá maximální velikost zprávy, kterou je schopen akceptovat.
- STARTTLS - umožňuje zašifrování spojení pomocí TLS [1], [9].

### 2.4.1 Ukázka spojení pomocí ESMTP

Tato praktická ukázka znázorňuje, jakým způsobem se liší začátek komunikace při zadání příkazu EHLO na základě kterého jsou klientovi (označován jako K) vypsány podporované příkazy ze strany serveru (označován jako S).

```
S: 220 mail.testmail.cz ESMTP Postfix (Debian/GNU)
K: EHLO ukazka
S: 250-mail.testmail.cz
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

# 3 Bezpečnost elektronické pošty

Jak již bylo uvedeno u jednotlivých protokolů elektronické pošty, problém elektronické pošty je v základu nezabezpečení komunikace. Nikomu tak není zabráněno odposlouchávat obsah samotných e-mailových zpráv, které jsou odesílány a přijímány. V horším případě při nezabezpečené komunikaci odchytne útočník i přihlašovací údaje včetně přihlašovacího hesla, což bude názorně předvedeno na příkladu.

Riziko odposlechnutí komunikace je vysoké zvláště v lokálních sítích. E-mail totiž cestou mezi adresátem a příjemcem cestuje mezi vícero servery a tak je i zde možnost jejího zachycení a následného přečtení.

Aby bylo tomuto možné zabránit, je nutné komunikaci mezi e-mailovým klientem a vzdáleným serverem nějakým způsobem zabezpečit. Obecně pro zabezpečení (šifrování) komunikace existují dva protokoly - SSL a TLS.

Dalším problémem nezabezpečené e-mailové komunikace je i to, že zprávy jsou často i po smazání z vlastní e-mailové schránky uloženy na serverech poskytovatele v zálohách. Také někteří poskytovatelé připojení mohou ukládat na své servery kopie doručovaných zpráv. Z těchto důvodů je možné zabezpečit e-mail i tak, že zašifrujeme samotnou e-mailovou zprávu.

## 3.1 Odchycení přihlašovacích údajů ke schránce

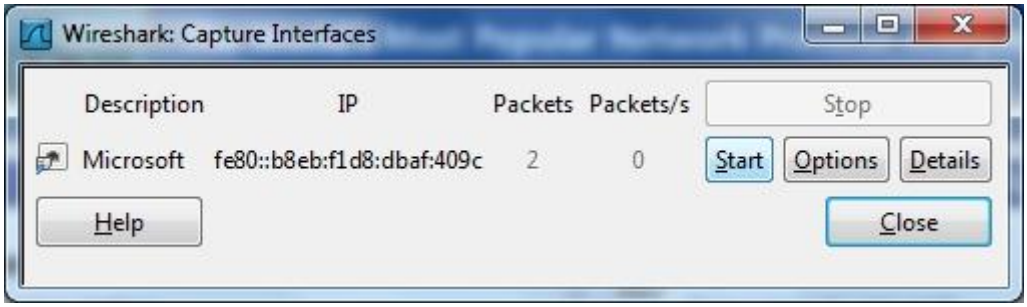
Pro pokus odchycení přihlašovacích údajů k vlastní schránce byl vybrán program Wireshark[4], který je velkým pomocníkem při zkoumání přenosu sítě. Za pomocí tohoto programu budou sledována přenášená data při přihlašování do e-mailové schránky na serveru Seznam.cz. Při přihlášení na tomto portálu je ve výchozím stavu zapnuté zabezpečené přihlášení za pomocí SSL. Jde zde ale i možnost tuto bezpečnostní funkci ručně vypnout. Její vypnutí zapříčiní přenášení přihlašovacích údajů v nezašifrované podobě, což je velmi vhodné pro účely této simulace.

Prvním krokem je přístup na adresu http://login.szn.cz/ a následné vynucení přihlašování bez použití SSL po kliknutí na příslušný odkaz vedle přihlašovacího okna. Tuto volbu je potřeba zvolit z důvodu přenosu přihlašovacích údajů v nešifrované podobě. Ve výchozím nastavení jsou přihlašovací údaje z bezpečnostních důvodů šifrovány.

[4] Program je ke stažení na oficiálních stránkách http://www.wireshark.org/ .


**Obrázek 3 - Přihlašovací stránka do e-mailové schránky**

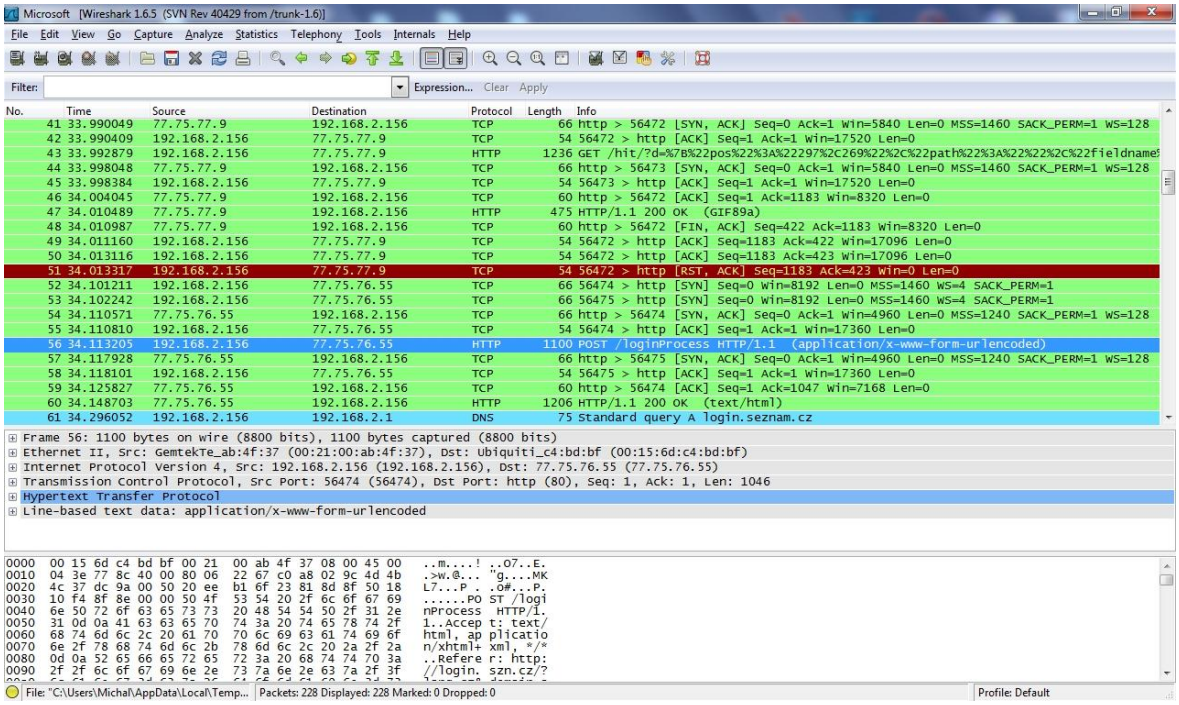
Dalším krokem ještě před samotným odesláním přihlašovacích údajů je spuštění programu Wireshark. Program funguje na principu sledování síťového provozu na zvoleném adaptéru. Proto je nutné po spuštění programu zvolit volbu „Capture Interfaces“. Po výběru této volby v hlavním menu programu se objeví okno, ve kterém je přehled dostupných síťových adaptérů, za pomocí nichž je možné sledovat síťový provoz. U testovacího počítače byl k dispozici jediný adaptér. Sledování síťové komunikace začne po stisku na tlačítko start u příslušného adaptéru.




**Obrázek 4 - Přehled dostupných rozhraní**

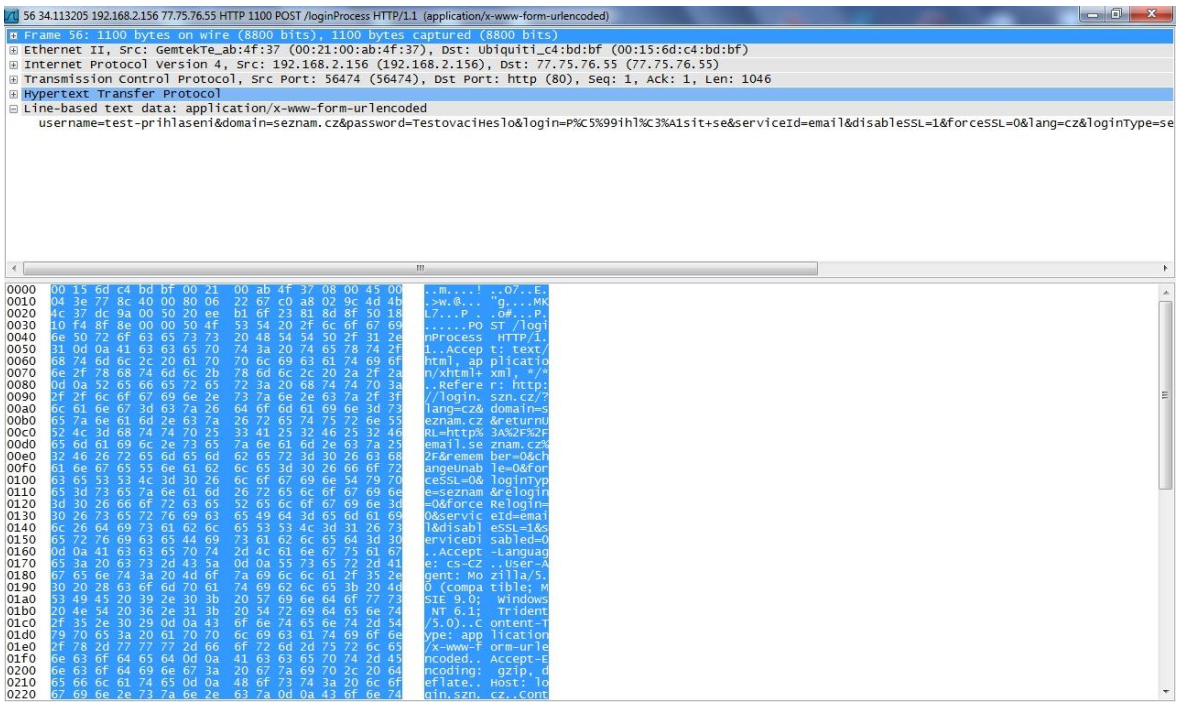
Po zvolení této volby se zobrazí hlavní okno programu Wireshark, ve kterém je již zobrazován zachytávaný síťový přenos.

V tento okamžik je již síťový přenos přes vydaný adaptér plně zachytáván, a tak je možné odeslat přihlašovací údaje do e-mailové schránky. Po přihlášení do schránky je již možné zastavit sledování síťového přenosu a ve výpisu přenosu najít příslušný paket. V tomto případě se bude jednat o paket na protokolu HTTP využívající POST.



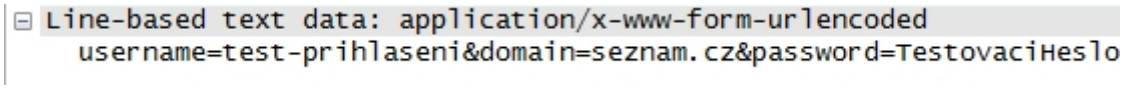

**Obrázek 5 - Zachytávané síťové přenosy**

Po nalezení správného paketu je možné zobrazit jeho detail, ve kterém bude možné vyčíst odeslané přihlašovací údaje. Detail paketu je zobrazen na následujícím obrázku.




**Obrázek 6 - Paket s přihlašovacími údaji**

V tomto detailu je tedy vidět kompletní obsah příslušného síťového paketu, který byl pomocí programu Wireshark zachycen. Hlavní pozornost je však nutné zaměřit na část Line-based text data, ve které je obsaženo přihlašovací jméno a heslo.




**Obrázek 7 - Odchycené přihlašovací jméno a heslo**

Jak je z obrázku patrné, tak přihlašovací jméno (username) je „test-prihlaseni“. Přihlašovací heslo (password) bylo pro účely pokusu nastaveno na řetězec „TestovaciHeslo“.

## 3.2 Šifrované spojení pomocí SSL a TLS

Protokol SSL (Secure Sockets Layer) je možností šifrování komunikace vyvinutou společností Netscape Communications. Tento protokol je vložen mezi transportní síťovou vrstvu a aplikační síťovou vrstvu [10].

První uvolněnou verzí protokolu SSL byla verze 2.0. Po níž následovala verze 3.0, která obsahovala zlepšení bezpečnostních vlastností. Další verzí je protokol TLS (Transport Layer Security), který je založen na specifikaci SSL 3.0 [10].

Veškerá přenášená komunikace je šifrována a je zajištěna autentizace komunikujících stran. Pro komunikaci je tak vytvořen uzavřený „tunel”, do kterého se útočník nedostane [10].

Protokol pracuje na principu asymetrických klíčů, kdy každý z účastníků komunikace má veřejný a soukromý klíč. Bližší postup výměny těchto klíčů a celkový postup komunikace je uveden zde [10].

1. Klient se spojí se serverem a sdělí vlastní podporované vlastnosti SSL a náhodně generovaná data.

2. Server jako odpověď zašle také tyto informace. K odpovědi připojí i svůj certifikát.

3. Klient ověří na základě certifikátu, zda jej vydala důvěryhodná certifikační autorita. Pokud ověření proběhlo úspěšně, tak klient vygeneruje základ klíče, kterým bude šifrována komunikace, ten zašifruje pomocí veřejného klíče serveru, který před chvílí spolu s certifikátem obdržel. To vše je zasláno serveru.

4. Server za pomoci svého soukromého klíče tuto komunikaci rozšifruje, získá tedy vlastní obsah komunikace, což byl základ šifrovacího klíče. Z tohoto základu šifrovacího klíče, který má aktuálně k dispozici klient i server, vygenerují oba hlavní šifrovací klíč.

5. Klient i server si po těchto výměnách potvrdí, že fáze výměn proběhla úspěšně a od tohoto okamžiku budou komunikovat za pomocí hlavního šifrovacího klíče. Tím končí fáze výměn klíčů (tzv. fáze handshake) a je vytvořeno šifrované spojení, pomocí kterého je nadále komunikováno [10].

## 3.3 Podvržení e-mailu

Jedním z problémů elektronické pošty je poměrně snadná možnost podvrhnutí odesílatele e-mailové zprávy. Často se stává, že uživatelům docházejí e-maily, ve kterých je banka žádá o zaslání přihlašovacích údajů po kliknutí na nějaký odkaz obsažený ve zprávě. Část příjemců zprávy plně věří poli odesílatel e-mailu, a tak se snadno dostane do problému.

Nyní bude názorně ukázáno, jak jednoduché toto podvrhnutí e-mailu opravdu je. Aby měl skript velmi jednoduché ovládání, tak k němu bude za pomocí HTML5 a CSS vytvořen jednoduchý formulář, přes který bude skript obsluhován. Koncový uživatel skriptu se tak nebude muset ani zajímat o jeho funkčnost, což ještě více zjednodušuje možnosti podvržení e-mailu.

Funkční část formuláře bude zajištěna za pomocí serverového skriptovacího jazyka PHP, který má pro účely odesílání elektronických zpráv vhodné prostředky. Konkrétně se jedná o funkci mail, díky které je odeslání zprávy velmi jednoduché.

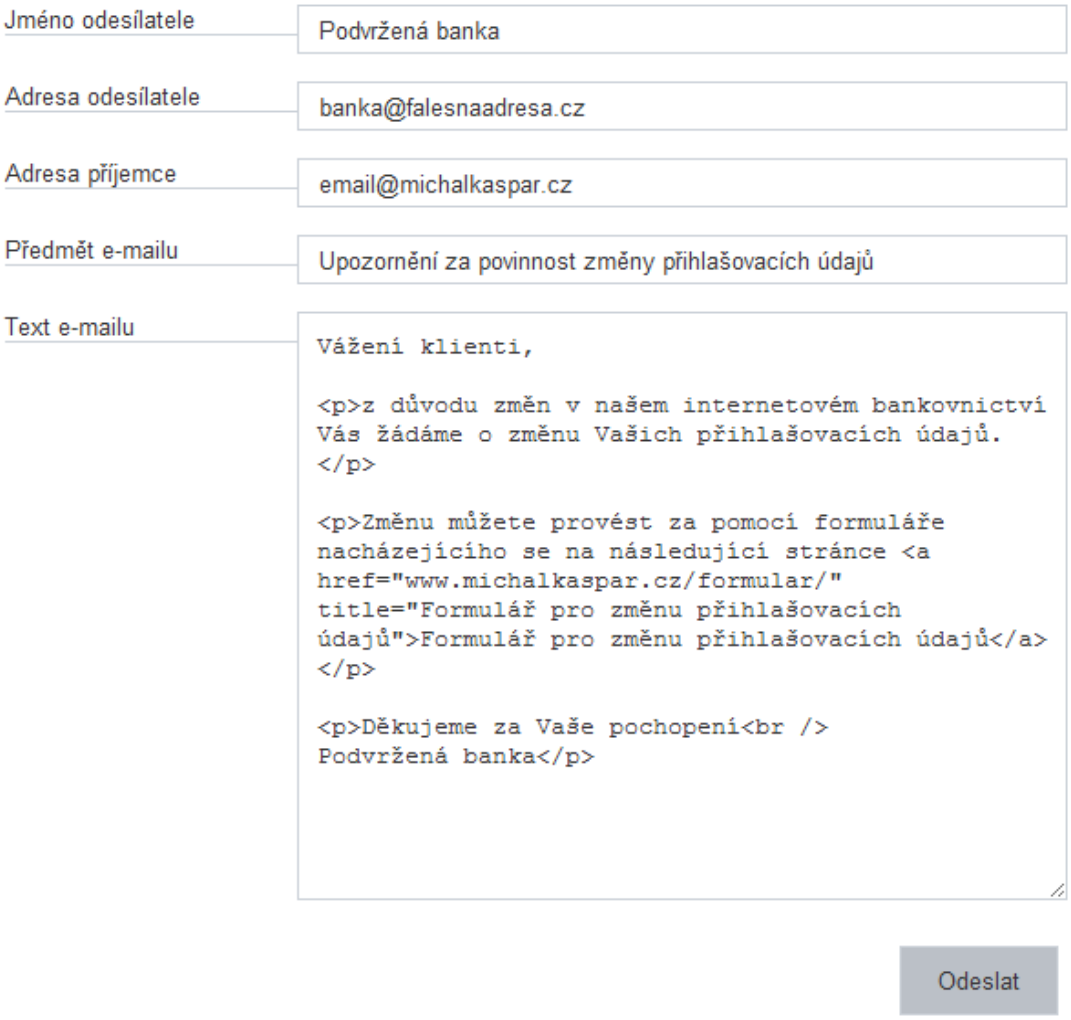
```
mail($adresaPrijemce, $predmetEmailu, $textEmailu, $header);
```

Jak je znázorněno na ukázce této funkce, obsahuje 4 parametry. Prvním parametrem je e-mailová adresa, na kterou má být zpráva odeslána. Jako druhý parametr je zadán předmět e-mailu a jako třetí samotný obsah. Posledním parametrem jsou takzvané hlavičky. Zde je možné použít libovolné kombinace hlaviček dle standardů elektronické pošty. Pro účely podvržení je tedy upravena hlavička FROM, ve které je podvrhem uvedena adresa například nějaké instituce.

Kompletní zdrojové kódy této praktické ukázky jsou uvedeny v přílohách A (soubor index.php), B (soubor Email.php) a C (soubor style.css).

Když je skript naprogramován a plně funkční, tak přichází čas pro praktické provedení této ukázky. Zasílání podobných zpráv za účely získání údajů od klientů je činností za hranicí zákona. Tudíž bude pouze pro účely ukázky zpráva odeslána na vlastní e-mailovou adresu.

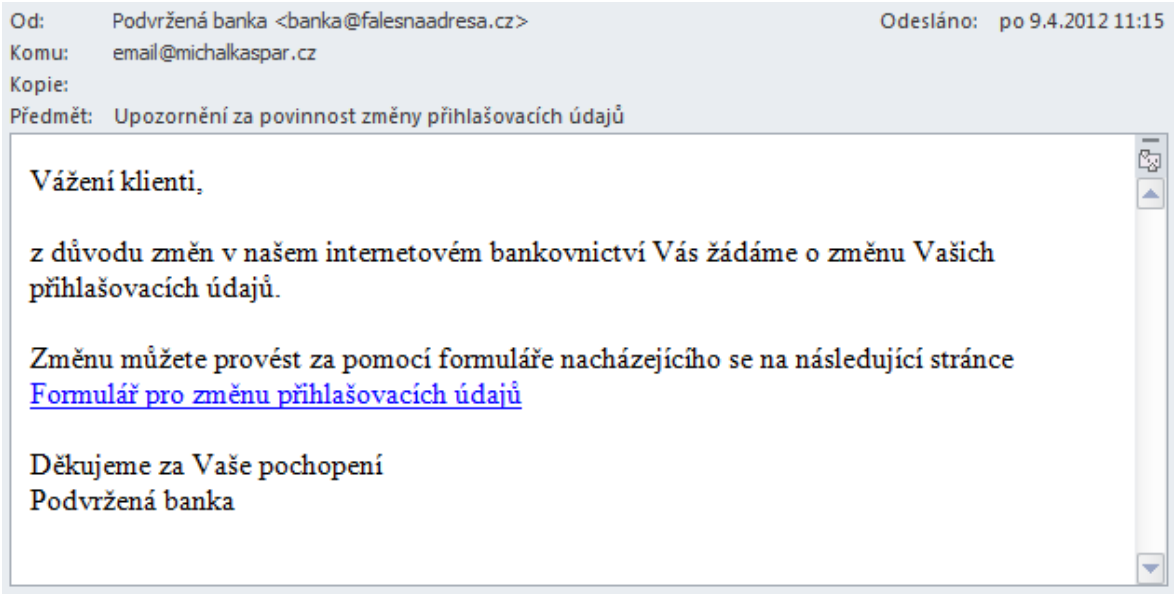
Pro ukázku bude zvolena možnost, která z příjemců láká přihlašovací údaje do internetového bankovnictví. Tyto údaje má klient dle přání banky vyplnit do speciálního formuláře, na který vede ze zprávy odkaz. Vytvořený skript odesílá zprávy ve formátu MIME. Tudíž je možné použít klasické prvky HTML jazyka. V tomto případě nejen formátování, ale i samotný odkaz. Vyplněný formulář připravený k odeslání vypadá následovně.




**Obrázek 8 - Formulář pro odeslání e-mailu**

Zpráva byla úspěšně odeslána a nyní je čas se podívat na výsledek, který dorazil. V e-mailovém klientovi zpráva nevypadá nijak podezřele. V poli odesílatele zprávy je zobrazeno, že e-mail dorazil z adresy skutečné banky, tudíž se část příjemců přestává bát a čte obsah zprávy.

Obsah vypadá také celkem věrohodně. Bylo by možné do něj doplnit ještě nějaký termín, po kterém by se internetové bankovnictví například zablokovalo. Uvedení termínu by zajisté ještě zvýšilo počet uživatelů, kteří by zprávu začali rychle a bez rozmýšlení řešit. Po kliknutí na odkaz se příjemce e-mailu dostává na stránku, která může na první pohled velmi jednoduše vypadat jako skutečné stránky banky. V případě vyplnění údajů do formuláře nastává velmi vážný problém.




**Obrázek 9 - Příchozí e-mail vytvořený PHP skriptem**

Na příkladu bylo naprosto základním způsobem ukázáno, jak velmi jednoduché je odeslat e-mail za někoho jiného. Přitom by bylo možné i samotnou e-mailovou zprávu naformátovat dle stylu, jakým zasílá banka například nějaké reklamní nabídky.

Na závěr ukázky je nutné podotknout, že bezmezně důvěřovat e-mailu nelze, a proto žádné instituce po klientech nežádají zasílání jakýchkoliv citlivých údajů na základě zaslání elektronické zprávy. I výpisy z účtu zasílané formou elektronické pošty vždy obsahují u zprávy přiložený e-mailový podpis. Právě tyto podpisy mají za úkol zaručit skutečného odesílatele zprávy, a proto se jim bude věnovat následující kapitola této práce

## 3.4 Elektronické digitální podpisy

Bylo nastíněno, že podvržení e-mailu je velmi jednoduché. Existuje však možnost, jak zaručit skutečného odesílatele. Touto možností jsou elektronické digitální podpisy e-mailu.

Elektronický digitální podpis (signatura) je nástroj, který umožní předat informaci o skutečném odesílateli elektronické pošty. Je možné tedy zajistit, že obsah zprávy nebyl mezi časem odeslání a přijetí změněn. Toto řešení využívají banky například při zasílání výpisu z účtu na e-mail. Druhou možností využití tohoto podpisu je zašifrování samotné zprávy [11].

Využívání elektronického podpisu je velmi jednoduché, stačí jeho nahrání do e-mailového klienta, kterého uživatel využívá a následně jeho připojení k e-mailové zprávě.

Ačkoliv existují i certifikační autority, které umožňují získání podpisu například na omezenou dobu zdarma, tak v rámci České republiky je často využívanou možností k získání elektronického digitálního podpisu Česká pošta. Ta nabízí zprostředkování elektronického podpisu vydávaného certifikační autoritou PostSignum QCA [12].

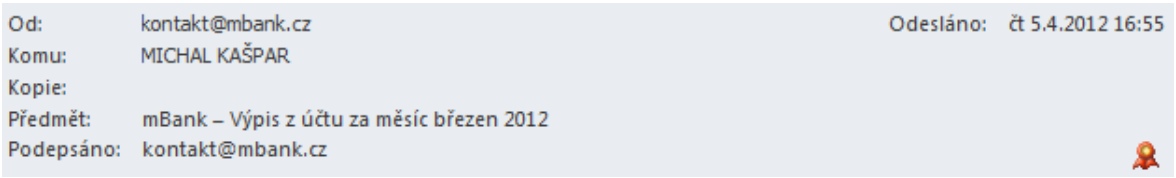
Pro získání touto cestou je nutné vyplnit příslušné formuláře a zaplatit poplatek za vystavení certifikátu[5].

Elektronické podpisy se zakupují s roční platností. Po roce je tedy u České pošty nutné zažádat o obnovení certifikátu[6].

Postup získání se může jevit sice jako zdlouhavý, ovšem v případech, kdy potřebujeme zaručit odesílatele e-mailu, je získání tohoto certifikátu nutností. Zajisté se nepředpokládá, že uživatelé budou podepisovat emaily zasílané přátelům. Ovšem v komunikaci s některými úřady je místo osobní návštěvy elektronický podpis možností ověření odesílatelovi totožnosti.

Patrně největší nevýhodou elektronických podpisů je fakt, že jsou vystavovány za poplatek, který není zrovna nejnižší. Pro člověka, který jej využije například dvakrát ročně při komunikaci s úřadem, se tak nemusí ani vyplatit.

Druhou menší nevýhodou pro uživatele, kteří neumějí příliš zacházet s počítačem je styl zobrazení podpisu. Tento fakt se velmi liší s použitým e-mailovým klientem, ale například v programu Microsoft Outlook 2010 je zobrazení podpisu následující.




**Obrázek 10 - Zobrazení podpisu v Microsoft Outlook 2010**

Jak je vidět, jedná se pouze o poslední řádek s informací podepsáno a ikonku na pravé straně zprávy, kterou lze poměrně lehce přehlédnout. Zůstává tedy otázkou, zda běžný uživatel tuto informaci ve zprávě vůbec registruje, natož aby ji sám aktivně vyhledával.

[5] Ceny za vydávané certifikáty - http://www.postsignum.cz/certifikaty.html.
[6] Obnova certifikátů PostSignum - http://www.postsignum.cz/obnova_certifikatu.html.

# 4 Nevyžádaná pošta

Do nevyžádané pošty dle nejčastějších definic patří hromadně rozesílané nabídky, které jsou většinou reklamního charakteru a mají za úkol nalákat příjemce zprávy k zakoupení nabízeného produktu. Právě z důvodu, že rozesílaní nevyžádaných zpráv přináší rozesílateli finanční zisk, je tento trend celosvětově velmi rozšířen [13].

Díky zmíněné definici nevyžádané pošty by se mohlo zdát, že vyvinout řešení pro její zastavení bude lehkou záležitostí. Opak je však do značné míry pravdou. Určit, co je nevyžádané a co nikoliv, je značně subjektivní záležitost [13].

Zajisté existuje určitý druh zpráv, který je nevyžádaný pro každého. Do této skupiny by se patrně daly bez problému zařadit e-maily oznamující vysokou výhru, zahraniční nabídky na zisk velkých peněz, reklamy na prodej léků, a podobně [13].

Horší situace však nastává například u rozeslané nabídky nějaké místní společnosti. Zde přichází na řadu subjektivní pocit každého příjemce e-mailu. Pro velkou část se nejspíše bude i v tomto případě jednat o nevyžádané obchodní sdělení. Pro zbytek se ale může jednat o zajímavou nabídku. Právě zde přichází první problém pro boj s nevyžádanou poštou, a to ten, že v případě velké skupiny lidí se její definice může značně lišit [13].

## 4.1 Původ slova spam

Nevyžádaná pošta je často označována jako spam. Kde se vzal tento výraz? V tomto bodě se některé zdroje poměrně rozcházejí. Tou nejrozšířenější verzí je ta, že název je inspirovaný značkou konzerv[7] vyráběných od 30. let minulého století dodnes. Tato konzerva byla hojně rozšířená během 2. světové války ve Velké Británii [13], [14].

Další proslavení pojmu spam jako něčeho nevyžádaného přinesla scénka ze seriálu Monty Python, kde si hosté chtějí objednat v restauraci. Všechny položky v jídelním lístku obsahují pojem spam [13], [14].

## 4.2 Příklady nevyžádané pošty

První zmínka o rozeslání spamu je uváděna ke květnu 1978, kdy zaměstnanec společnosti Digital Equipment Corporation po tehdejší síti ARPANET rozeslal firemní prezentaci [14].

Mezi první nejvýraznější události v oblasti nevyžádané pošty v rámci České republiky patří hromadný spam odeslaný společností Media Online, s. r. o. Tato společnost, v té době provozující internetový postál tvujdum.cz, se dopustila rozeslání spamu. V něm informovala o novinkách na svém webu [15], [16].

Rozeslání zprávy mělo samozřejmě velkou odezvu a nenechala na sebe dlouho čekat i reakce příjemců zprávy. Dle serveru Lupa.cz tehdy 30 000 lidí zaslalo stížnost na tuto společnost. Za rozeslání e-mailu byla společnosti udělena pokuta, jejíž výše není přesně

[7] Stránky společnosti Hormel Foods Corporation - http://www.spam.com.

známá. Ředitel této společnosti taktéž uznal jisté pochybení a jako akt odpovědnosti následně věnoval 50 000Kč nadaci Člověk v tísni [16].

## 4.3 Algoritmy a způsoby ochrany proti nevyžádané poště

Celosvětové rozšíření nevyžádané pošty je důvod pro neustálý vývoj velkého množství různých způsobů její detekce a snahy ji potlačit. Může se jednat o základní kontroly samotného obsahu e-mailové zprávy až po srovnávání odesílatele s centrální databází. Díky neustálému vývoji na straně rozesílatelů je nutné tyto principy kontroly neustále aktualizovat a upravovat. Jelikož je rozesílání těchto reklamních zpráv pro spoustu subjektů poměrně výnosným byznysem, tak se dá i do budoucna očekávat pokračování tohoto problému, na který se bude muset i nadále reagovat vytvářením stále účinnějších algoritmů na detekci [6].

### 4.3.1 Základní analýza obsahu e-mailu

Základní analýza obsahu e-mailu patří mezi jednu z nejjednodušších variant kontroly e-mailových zpráv na nevyžádanou poštu. Kontrola vychází z faktu, že nevyžádaná pošta je z velké části propagací nějakého produktu. V e-mailových zprávách jsou tedy vyhledávány slova a sousloví, která se objevují v nevyžádané poště velmi často. Typickým příkladem slova, které tyto způsoby kontroly zprávy zachytí, je slovo viagra [6], [17], [18].

Rozesílatelé nevyžádané pošty samozřejmě na tyto kontrolní mechanismy zareagovali a dochází tak k různému deformování klíčových slov a úpravám. Často je vidět nahrazení písmene „a“ za pomocí symbolu „@“. Novější úpravy těchto kontrolních mechanismů však již pracují s různými regulérními výrazy a dokáží odhalit i takto upravované zprávy [6], [17], [18].

### 4.3.2 Black list

V případě black listů se jedná o centrální distribuované seznamy, které obsahuje informace o IP adresách či DNS záznamech, ze kterých již v minulosti byla rozesílána nevyžádaná pošta. Existuje velké množství black listů ať poskytovaných zdarma, či komerčně [6], [17], [18].

Jelikož jsou servery poskytující tyto seznamy často pod vlnou různých útoků, tak jsou velmi často poskytovány za pomocí zrcadel rozmístěných různě po světě. Díky tomuto řešení jsou odolnější například proti DoS útokům [6], [17], [18].

Problém, který black listy provází již od počátků jejich fungování, jsou případy, kdy se na seznam dostane omylem i nesprávný záznam. V takovém případě může být korektně odeslaná zpráva označena jako nevyžádaná. Samozřejmě existuje možnost požádat o odstranění z takového seznamu, ale jedná se o časově zdlouhavou záležitost, během které mohou být problémy s falešným označováním zpráv [6], [17], [18].

Dalším problémem je fakt, že poměrně často jsou pro rozesílání pošty zneužity zavirované počítače běžných uživatelů, tudíž i doposud neznámých IP adres [6], [17], [18].

### 4.3.3 White list

White list je do jisté míry opakem dříve uvedeného black listu. Nejedná se v tomto případě o centrálně distribuovaný seznam, ale o lokální seznam e-mailových adres. Adresy v tomto seznamu nebudou testovány na přítomnost nevyžádané pošty [6], [17], [18].

Toto řešení je často využívané v případech, kdy žádané zprávy končí označené jako nevyžádané. Může se jednat například o pravidelně generované a zasílané reporty z nějaké aplikace, které uživatel potřebuje doručit [6], [17], [18].

Do tohoto seznamu jsou tedy přidávaný adresy, které jsou pro uživatele věrohodné, nebo s nimi často komunikuje, a tak není potřebné takové zprávy kontrolovat na nevyžádanou poštu [6], [17], [18].

### 4.3.4 Grey list

Takzvaný grey listing je další metodou k zabránění příchodu nevyžádané pošty. Tato metoda využívá faktu, že rozesílatelům nevyžádané pošty jde zejména o rychlost. Rozesílají za pomoci robotů miliony zpráv, a tak je pro ně určité procento nedoručených zanedbatelné [6], [17], [18], [19], [20].

Při práci s grey listem je kontrolována pouze IP adresa odesílatele, e-mailová adresa odesílatele (v rámci SMTP) a e-mailová adresa příjemce (v rámci SMTP). V tomto případě tedy není žádným způsobem kontrolovaný samotný obsah e-mailu [6], [17], [18], [19], [20].

Tato metoda pracuje již na úrovni MTA a princip jejího fungování je takový, že příchozí zpráva je v případě nového odesílatele, se kterým ještě nebylo komunikováno, odmítnuta a odesílateli je odeslána například následující chyba [6], [17], [18], [19], [20].

```
450 Requested mail action not taken: mailbox unavailable [20].
```

Tato chyba znamená, že je schránka dočasně nedostupná. Servery běžných uživatelů, které respektují příslušné RFC, se o odeslání zprávy pokusí po nějakém časovém intervalu znovu [6], [17], [18], [19], [20].

V případě rozesílatelů nevyžádané pošty jsou využíváni roboti a ohromné databáze e-mailových adres, tudíž nefunkční přijetí u nějakého adresáta není většinou řešeno. Z tohoto pohledu se tato metoda jeví tedy jako velmi účinná [6], [17], [18], [19], [20].

Výhodou této metody je její nenáročnost. Na prostředky e-mailového serveru je mnohem méně náročná než například složité algoritmy kontroly samotného obsahu zprávy. V případě kombinace s dalšími metodami se může jednat o účinný první stupeň kontroly pošty, na který může navazovat například kontrola samotného obsahu e-mailu [6], [17], [18], [19], [20].

Naopak nevýhodou je zdržení doručení pošty. Než se server odesílatele pokusí zprávu odeslat znovu, může uběhnout čas v řádech minut až hodin. V případě špatně

nakonfigurovaného serveru odesílatele také nemusí být zpráva již znova odeslána. V případě pravidelné komunikace se však jedná pouze o zdržení v doručování první zprávy [6], [17], [18], [19], [20].

### 4.3.5 Bayesovský antispamový filtr

Bayesovské antispamové filtry jsou již řešením složitějším, kde je kontrola obsahu zprávy již na vyšší úrovni. Pracují stále na základním principu, že slova používané v nevyžádané poště se opakují a budou použita i v dalších nevyžádaných zprávách. Díky tomuto faktu jsou slova a jednotlivé části e-mailové zprávy porovnávány s databází vzorků nevyžádané pošty. Na základě počtu pozitivních mezivýsledků je vyhodnocována pravděpodobnost, že je zpráva nevyžádaná [6], [17], [18], [21].

Kromě kontroly samotných slov je v případě těchto filtrů vyhodnocováno i použití diakritiky a velkých písmen v obsahu zprávy. Filtr tedy zareaguje například výrazněji na výraz „SLEVA“ než na výraz „sleva“. Taktéž použití několika „!“ může zvýšit pravděpodobnost toho, že je zpráva nevyžádanou [6], [17], [18], [21].

Dalším faktorem kontroly jsou například externí URL odkazy a jejich obsah, ale taktéž i použití HTML značek v e-mailu. Díky množství vyhodnocovaných parametrů mají tyto filtry v případě dostatečné databáze poměrně velkou účinnost [6], [17], [18], [21].

### 4.3.6 Další způsoby boje proti nevyžádané poště

Výčet uvedených metod pro boj s nevyžádanou poštou není samozřejmě kompletní. Spíše se jedná o ukázku těch nejzákladnějších. Existuje velké množství dalších metod používaných pro boj s nevyžádanou poštou [6], [17], [18].

Jako příklad dalšího způsobu boje s nevyžádanou poštou je možno uvést metodu porovnávání signatur zpráv. Tato metoda využívá faktu, že v případě velkého množství zpráv se stejnou signaturou se bude pravděpodobně jednat o nějaké nevyžádané sdělení [6], [17], [18].

Tato metoda je však do jisté míry obcházena částečným dynamickým generováním obsahu e-mailu. Jejich signatura se pak liší. Jako reakce na toto obcházení bylo zavedeno porovnávání částí jednotlivých zpráv a sčítání počtu identických částí [6], [17], [18].

# 5 Průzkum mezi poskytovateli e-mailových služeb

Pro získání aktuálních údajů z roku 2012 byl vypracován vlastní orientační průzkum mezi několika poskytovateli e-mailových služeb. Otázky byly zaměřené na používané technologie pro zajištění služeb a také na některé statistiky týkající se nevyžádané pošty.

### 5.1.1 Seznam účastníků průzkumu

Průzkumu se zúčastnilo několik společností, které odpovídaly na několik otázek formou internetového dotazníku. Jedná se o společnosti nabízející kompletní služby v oblasti webhostingu, ale i společnosti nabízející e-mailová řešení zdarma. Kompletní přehled společností zapojených do průzkumu je uveden v následující tabulce.

**Tabulka 1 - Společnosti zapojené do průzkumu**

| Název společnosti | Webové stránky |
|---|---|
| EBOLA Czech s. r. o. | ebola.cz |
| WEBYA hosting, s. r. o. | webya.cz |
| Economia, a. s. | volny.cz |
| HostingSolutions s. r. o. | hostingsolutions.cz |
| Seznam.cz a. s. | seznam.cz |
| savana.cz s. r. o. | savana.cz |
| WEDOS Internet | wedos.org |

### 5.1.2 Používaný operační systém na e-mailových serverech

Prvním parametrem průzkumu byl použitý operační systém v rámci serverů sloužících pro e-mailové účely a dále zjištění jeho verze. V případě různých verzí, alespoň přibližný údaj.

**Tabulka 2 - Používané operační systémy**

| Název společnosti | Operační systém | Verze operačního systému |
|---|---|---|
| EBOLA Czech s. r. o. | Linux | Gentoo |
| WEBYA hosting, s. r. o. | Linux | CentOS release 5.7 (x86_64) |
| Economia, a. s. | Linux | CentOS 6.1 |
| HostingSolutions s. r. o. | Linux | CentOS, kernel 2.6 a vyšší |
| Seznam.cz a. s. | Linux | Debian Squeeze |
| savana.cz s. r. o. | Linux | - |
| WEDOS Internet | Linux | CentOS |

Jak je z tabulky patrné, tak pro účely e-mailových serverů byl ve společnostech zapojených do průzkumu použitý vždy Linux. V rámci distribucí je nejčastěji používanou distribucí operační systém CentOS. Mezi další používané distribuce patřil Debian a Gentoo.

### 5.1.3 Používané e-mailové a antispamové řešení

Druhým parametrem průzkumu bylo zjištění, jaký konkrétní e-mailový systém je na serverech používaný. Další otázkou bylo, jaké antispamové metody jsou použité. Z přehledu v tabulce je patrné, že zde se již používané technologie značně liší. V rámci

používaného e-mailového řešení se zde objevují open-source řešení v podobě Postfix, Exim a Qmail. Nechybí zde ani komerční řešení v podobě IceWarp.

Z použitých antispamových řešení jsou shodné rysy v podobě používaní různých seznamů (black list, white list, grey list). Dále zde dochází ke shodě v používání open-source produktu SpamAssassin.

**Tabulka 3 - Používané e-mailové a antispamové systémy**

| Název společnosti | E-mailový systém | Antispamový systém |
|---|---|---|
| EBOLA Czech s. r. o. | Postfix | SpamAssassin, blacklisty, fuzzyOCR, greylisting |
| WEBYA hosting, s. r. o. | Exim 4.7x | SpamAssassin |
| Economia, a. s. | - | Kaspersky, SpamAssassin |
| HostingSolutions s. r. o. | Qmail | SpamAssassin, g/b/w lists, domainkeys, dnsbl |
| Seznam.cz a. s. | Vlastní řešení | Vlastní řešení |
| savana.cz s. r. o. | IceWarp | IceWarp |
| WEDOS Internet | IceWarp | SpamAssassin |

### 5.1.4 Statistiky e-mailů

Poslední částí miniprůzkumu byla snaha o vytvoření aktuálních statistik s přehledem o počtu přijatých a odeslaných e-mailů přes servery daných společností. V neposlední řadě bylo v tomto průzkumu taktéž zjišťováno, jak moc velkým problémem nevyžádané pošta je.

**Tabulka 4 - Počty e-mailů a podíl nevyžádané pošty**

| Název společnosti | Přijaté/den | Odeslané/den | Přijaté SPAMy/den |
|---|---|---|---|
| Economia, a. s. | 1 400 000 | 800 000 | 300 000 |
| HostingSolutions s. r. o.[8] | 3 200 000 | 1 500 000 | 2 400 000 |
| Seznam.cz a. s. | 20 000 000 | 5 000 000 | 97 % na příchodu |
| savana.cz s. r. o. | 1 500 000 | 800 000 | 90 % všech emailu |

Jak je z průzkumu patrné, nevyžádané pošta je obrovským problémem. Například u společnosti Seznam.cz, která je patrně největším českým poskytovatelem e-mailové schránky zdarma, je počet přijatých e-mailů za den roven průměrně 20 000 000. Dle dodaných údajů od mluvčí společnosti je množství nevyžádané pošty rovno 97 % příchozích e-mailů, což je zajisté pozoruhodné číslo. Právě podobná čísla přinášejí potřebu boje proti nevyžádané poště.

[8] Statistické údaje jsou jen ze serverů, kde probíhá filtrace zpráv skrze SpamAssassin.

# 6 Instalace a konfigurace e-mailového serveru

Tato kapitola bude zaměřena na instalaci a konfiguraci emailového serveru. Veškerá konfigurace bude probíhat na virtuálním serveru poskytnutém společností WEDOS, a. s. pro účely vypracování konfigurace na reálném zázemí. K serveru byla zakoupena česká doména testmail.cz, na níž bude znázorňována konfigurace.

## 6.1 Software použitý na straně serveru

Jako distribuce pro e-mailový server byla díky předchozím zkušenostem se systémem Ubuntu zvolena instalace Debianu. Instalované balíky budou použity z repozitářů Debianu. Níže je uveden přehled hlavních softwarových produktů instalovaných na serveru.

- Debian Squeeze[9] (6.0.4 - 64bit) - operační systém instalovaný na virtuálním serveru.
- Postfix[10] - odesílání a přijímání e-mailů ze sítě Internet.
- Dovecot[11] - ukládání e-mailů na disk a zpřístupňování uživatelům za pomocí protokolů POP a IMAP.
- AMaViS[12] - kontrola e-mailů ve spolupráci s antivirem a antispamem.
- Clam Antivirus[13] - antivirové řešení pro kontrolu e-mailových zpráv.
- SpamAssassin[14] - antispamové řešení pro kontrolu e-mailových zpráv.
- SquirrelMail[15] - online e-mailový klient pro uživatelský přístup ke zprávám.

Dále budou instalovány různé doplňkové softwarové balíky, které jsou potřebné pro funkčnost dříve uvedených. Bude tak například instalován webový server Apache2, generátor pro SSL certifikáty a další.

Každý z uvedených balíků bude mít jasnou funkci při zpracování, či přenosu e-mailové zprávy. Postup zpracování jde rozčlenit do několika kroků.

V prvním kroku při příchodu e-mailu pomocí protokolu SMTP převezme e-mail Postfix a provede základní zpracování. V tomto kroku je zjištěno, byl-li odesílatel přihlášený nebo je-li příjemce e-mailu uživatelem v systému. Dále je možné mít nastavenou kontrolu na základně různých listů nevyžádaných odesílatelů. Na základě těchto základních zpracování Postfix rozhoduje, jestli zprávu přijímá k dalšímu pracování, nebo je odmítne [22].

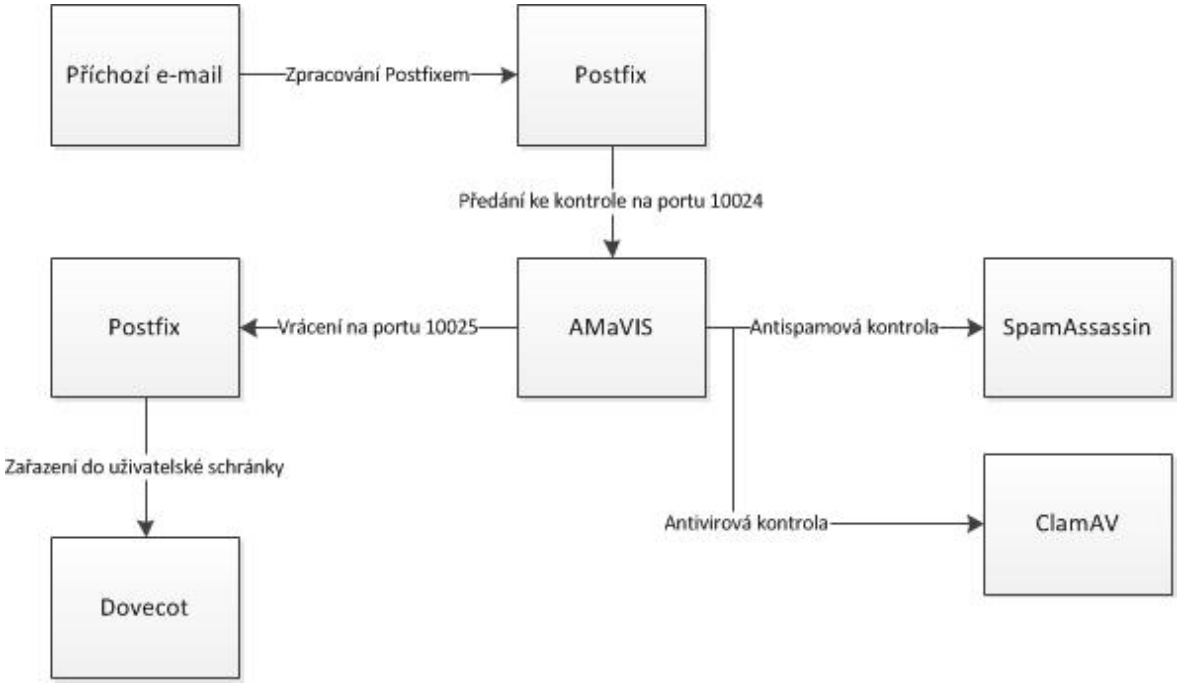
Dalším krokem je předání zprávy ke zpracování AMaViSem pro účely kontroly obsahu. Předání je provedeno standardně na TCP portu 10024. Je provedena kontrola za pomocí

[9] http://www.debian.org/
[10] http://www.postfix.org/
[11] http://www.dovecot.org/
[12] http://www.amavis.org/
[13] http://www.clamav.net/lang/en/
[14] http://spamassassin.apache.org/
[15] http://squirrelmail.org/

produktu SpamAssassin, který provádí kontrolu na přítomnost nevyžádané pošty. Každá zpráva je tímto produktem ohodnocena. Pokud je překročena nastavené hranice, je označena jako SPAM pomocí doplnění e-mailové hlavičky a případnou změnou předmětu e-mailu. Poté následuje kontrola zprávy na přítomnost virů za pomoci antivirového programu ClamAV [22].

Po tomto kroku následuje vrácení zprávy Postfixu, které je standardně provedené za pomocí TCP postu 10025. Nakonfigurovaný Postfix považuje zprávy přijaté na tomto portu za zkontrolované, tudíž je neposílá zpět AMaViSu [22].

Posledním krokem je předání zprávy Dovecotu, který ji uloží do adresáře příslušného uživatele, aby k ní následně uživateli za pomoci protokolů POP a IMAP umožnil přístup [22].




**Obrázek 11 - Schéma použitých součástí e-mailového serveru**

## 6.2 Nasměrování domény na VPS

Pro účely vypracování této práce v reálných možnostech byla zakoupena česká doména testmail.cz. Aby bylo možné s ní ve spolupráci se serverem nějak pracovat, je nutné ji pomocí DNS na tento server nasměrovat.

Doména bude využívat DNS servery společnosti WEDOS, a. s., tudíž se tato změna provede v administraci. Bude nutné nastavit dva typy záznamů. Prvními z nich jsou takzvané a záznamy, které budou potřebné pro případ přístupu do online e-mailového klienta. Zde bude nastavena IP adresu serveru, která je 31.31.78.176. Druhým typem jsou takzvané MX záznamy, které budou využity pro samotnou funkčnost e-mailového serveru.

Zde bude zadána adresa mail.testmail.cz, která bude poté využívána pro přístup k e-mailovým serverům. Například pro přístup pomocí protokolu POP, IMAP, či SMTP.

Na následujícím obrázku je ukázáno výsledné nastavení DNS. Nyní je již adresa správně propojena se serverem a je tedy možné pokračovat dále.

| | název | TTL | typ | data |
|---|---|---|---|---|
| | | 1800 | A | 31.31.78.176 |
| | | 1800 | MX | 1 mail.testmail.cz |
| | * | 1800 | A | 31.31.78.176 |

**Obrázek 12 - Nastavení záznamů u domény testmail.cz**

## 6.3 Instalace operačního systému Debian Squeeze

Pro instalaci je nejprve nutné v zákaznické administraci k mechanice serveru připojit instalační médium operačního systému a naplánovat restart serveru s připojeným instalačním médiem. V tomto případě bude připojeno instalační médium systému Debian 6.0.4 v 64bitové variantě.

Následně je možné se k serveru připojit za pomocí přihlašovacích údajů ke KVM obdržených od poskytovatele. K tomuto připojení je možné využít například program VeNCrypt[16].

Nyní již začíná samotná část instalace, která spočívá v jednoduchém následování instalačních obrazovek. Celá instalace obnáší průchod zhruba třiceti instalačními obrazovkami.

Za zmínku stojí hlavní položky nastavené při instalaci systému. Mezi tyto nastavení patří:

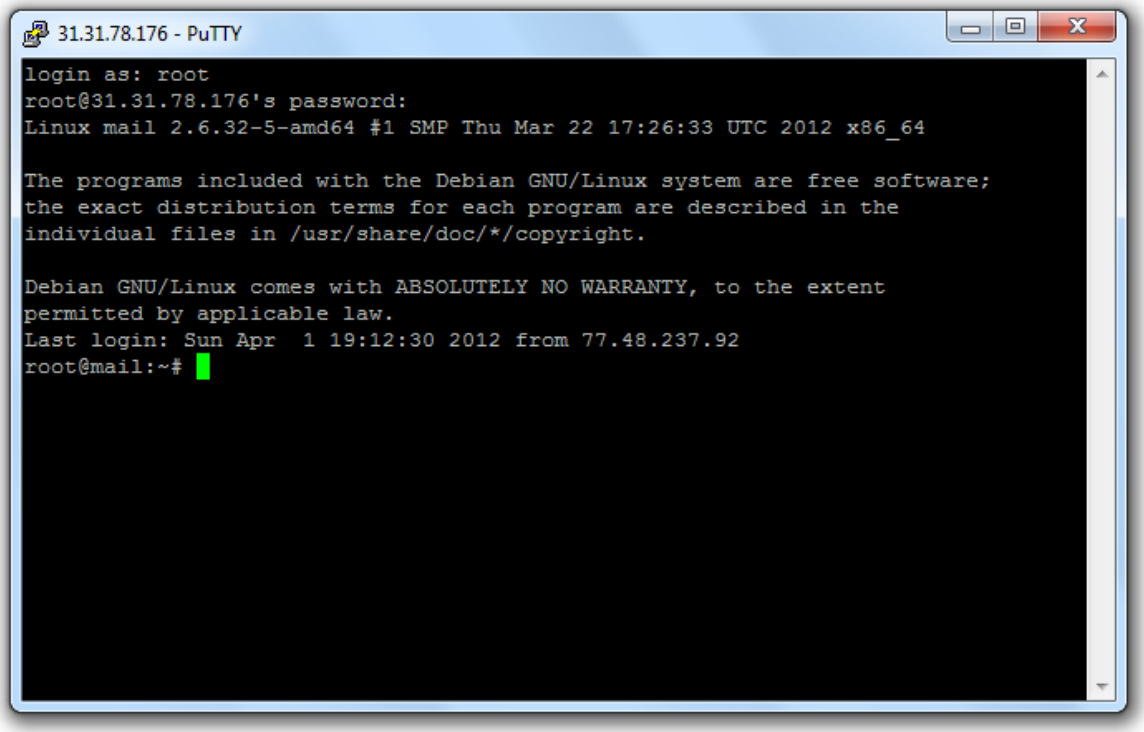
- IP adresa: 31.31.78.176
- Maska: 255.255.255.0
- Brána: 31.31.78.1
- DNS servery: 46.28.104.68 a 31.31.72.3
- Hostname: mail
- Domain name: testmail.cz

Jako další je při instalaci nastaveno heslo pro správce systému a vytvořen další uživatelský účet. Po dokončení instalace a restartu serveru je ještě nutné nainstalovat SSH server. Instalace je provedena za pomocí příkazu

```
apt-get install ssh
```

Tímto je instalace dokončena a na další přístup k serveru bude využit již SSH klient (například PuTTY).

[16] Program VeNCrypt je ke stažení na oficiálních stránkách - http://sourceforge.net/projects/vencrypt/.




**Obrázek 13 - Připojení k SSH za pomocí programu PuTTY**

## 6.4 Instalace Postfix

Tato podkapitola s instalací Postfixu je zpracována převážně podle zdrojů [23], [24], [25], [26], [27], [28], [29].

Nyní je na serveru připravený funkční operační systém a je zpřístupněna možnost přístupu pomocí SSH. Na řadu tedy přichází samotná instalace e-mailového serveru. Bude instalována kombinace programů Postfix a Dovecot. Pro účely bezpečnosti a taktéž zabránění zneužití serveru k rozesílání nevyžádané pošty bude zároveň nastavena autorizace SMTP serveru.

Pro začátek je nutné za pomocí balíčkovacího systému získat balíky potřebné k instalaci a následné konfiguraci. To se vykoná zadáním příslušného příkazu.

```
apt-get install postfix sasl2-bin libsasl2-2 libsasl2-modules
```

Po zadání příkazu je ještě nutné samotnou instalaci odsouhlasit. Při instalaci je uživatel požádán o zadání system mail name. To je použito například při odeslání e-mailu z neúplné adresy.

Po dokončení samotné instalace přichází řada na editaci konfiguračního souboru programu Postfix (/etc/postfix/main.cf). Tento soubor bude z části upraven a doplněn pro specifická nastavení e-mailového serveru.

Veškeré úpravy textových souborů budou prováděné editorem nano. Je tedy nutné otevřít konfigurační soubor /etc/postfix/main.cf za pomocí následujícího příkazu.

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

Po zadání příkazu a otevření konfiguračního souboru v editoru je na řadě samotná editace.

Na začátek souboru bude umístěno několik základních nastavení o doméně, pojmenování serveru a destinacích.

```
myhostname = mail.testmail.cz
mydomain = testmail.cz
myorigin = $mydomain
mydestination = $mydomain, $myhostname, localhost.$mydomain, localhost
```

První řádek myhostname obsahuje nastavené jméno poštovního serveru. V druhém parametru pojmenovaném mydomain je nastavená doména, která bude v provozu využívána.

Třetím parametrem řeší případy, kdy je ze serveru odeslána zpráva s neúplnou adresou. Jedná se například o zprávu vytvořenou za pomoci plánovače CRON a odeslaná pod uživatelem root. Ve výchozím nastavení by se do adresy doplnil parametr myhostname. Zpráva by tak byla odeslána v tomto případě z adresy root@mail.testmail.cz. Pro potlačení tohoto chování slouží právě parametr myorigin. Zde je přiřazeno nastavení mydomain, aby byla doplňována pouze doména.

Poslední výše uvedená položka mydestination obsahuje informace o doménách, pro které má Postfix přijímat poštu. V tomto případě je žádoucí, aby přijímal poštu pro nastavené údaje v položkách myhostname a mydomain. Z výchozí konfigurace budou ponechány dvě varianty s localhost.

Další částí konfiguračního souboru je blok, který byl automaticky vygenerovaný přímo Postfixem a bude v souboru dále zanechán. V tomto bloku se nacházejí informace o aliasech, nastavení banneru zobrazovaného po připojení k serveru či nastavení limitu schránky.

```
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
append_dot_mydomain = no
delay_warning_time = 4h
readme_directory = no
```

Další následně doplněnou konfigurací je nastavení umístění e-mailových schránek. Tato konfigurace serveru bude využívat přihlašování pomocí uživatelských účtů přímo

ze systému Debian. Nabízí se tedy možnost ukládat zprávy do složky Maildir v uživatelské domovské složce. Tato informace bude také nastavena i v konfiguračním souboru programu Dovecot.

home_mailbox = Maildir/

V konfiguraci následuje blok nastavení pro zabezpečené spojení pomocí TLS. Jedná se o výchozí nastavení, která nesou informace o podpoře TLS a použitých certifikátech. V základním nastavení je zde nastaven certifikát poskytnutý se samotným Postfixem. V případě reálného nasazení je dobré jej nahradit vlastním certifikátem od ověřené certifikační autority. Pro účely ukázky bude plně postačující tento výchozí.

```
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
```

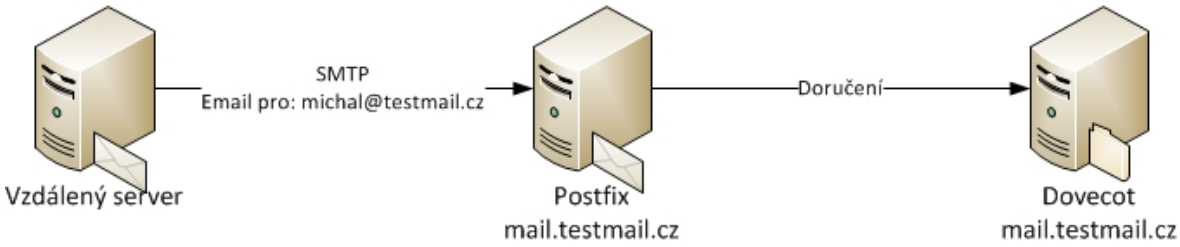
Pro zvýšení bezpečnosti e-mailového serveru a taktéž pro zabránění jeho zneužití pro účely rozesílání nevyžádané pošty bude doplněna konfigurace sloužící k autentizaci klienta u serveru. Neautentizovaný klient tedy nebude moci zneužít server pro rozesílání zpráv na ostatní servery. V dnešní době se jedná již o jakýsi bezpečnostní standard, bez kterého by mohl být server poměrně rychle zneužit.

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
broken_sasl_auth_clients = yes
```

První řádka konfigurace smtpd_sasl_auth_enable slouží k zapnutí autentizace, druhá smtpd_sasl_type nastavuje využívání SASL přihlašovaní na místo výchozí Cyrus SASL. Konfigurace smtpd_sasl_path určuje, kde Postfix nalezne autentizační server Dovecotu. Další řádka konfigurace smtpd_sasl_security_options zajistí skutečné ověření klientových autentizačních údajů. Zakáže se tak anonymní autentizace SMTP, kterou je možné zneužít pro rozesílání nevyžádané pošty. U položky smtpd_sasl_local_domain je nastaven řetězec, který je doplněn k neúplnému přihlašovacímu jménu. Poslední řádek konfigurace bloken_sasl_auth_clients umožňuje alternativní zápis určený nestandartním klientům u kterých jinak SMTP AUTH nefunguje. Mezi nestandartní klienty patří například starší verze Microsoft Outlook.

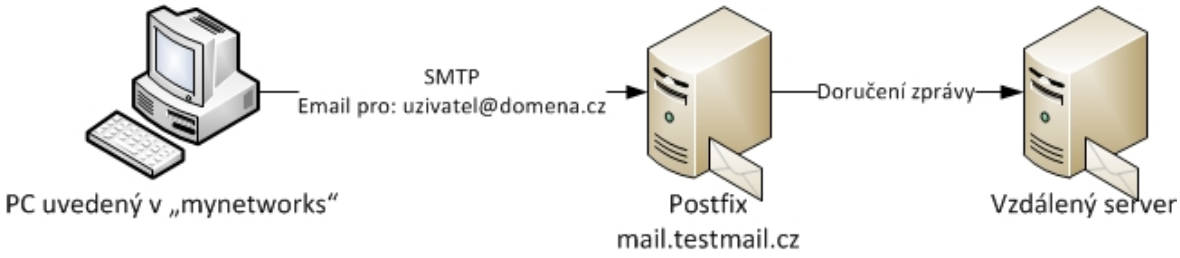
Dalším doplněnou konfigurací je část s omezeními. Existuje několik variant, kdy je průchod zprávy žádoucí a je potřeba jej povolit. V ostatních příkazech zpráva nebude zpracována.

První případ znázorňuje odeslání e-mailu ze vzdáleného serveru na lokální. Od odesílatele v tomto případě pochopitelně nelze vyžadovat přihlášení. Zpráva bude převzata k doručení pouze v případě, že příjemce je uživatelem lokálního systému.



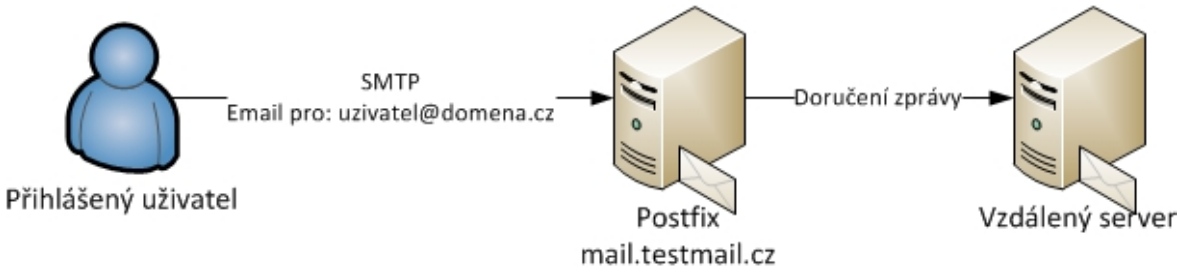

**Obrázek 14 - První žádoucí způsob průchodu emailu**

Druhým případem je odesílání zprávy z lokálního systému na vzdálený bez potřeby přihlášení. Jedná se o případ, kdy je odesíláno z lokální sítě. Uživatel je tak uveden v sekci mynetworks.




**Obrázek 15 - Druhý žádoucí způsob průchodu emailu**

Třetím případem povoleného odeslání je případ, kdy uživatel lokálního systému provede přihlášení k serveru. Poté server zprávu odešle i na vzdálený e-emailový server. Jelikož došlo k přihlášení, tak není důvod odeslání zprávy blokovat.




**Obrázek 16 - Třetí žádoucí způsob průchodu emailu**

Oproti tomu poslední obrázek znázorňuje situaci, kterou je nutné zablokovat z důvodu pravděpodobného zneužití. V tomto případě se prostřednictvím lokálního serveru snaží nepřihlášený uživatel odeslat zprávu na vzdálený server. Tato zpráva nebude odeslána.


**Obrázek 17 - Nežádoucí způsob průchodu emailu**

Právě dodržování uvedených podmínek bude nastaveno v následujících sekcích. Sekce smtpd_recipient_restrictions je vyhodnocována v okamžik, kdy klient serveru v SMTP spojení odešle příkaz RCPT TO. Následně se vyhodnocují jednotlivá nastavení.

```
smtpd_recipient_restrictions =
  permit_mynetworks,
  permit_sasl_authenticated,
  reject_unauth_destination,
```

Toto nastavení tedy splňuje podmínky předchozích příkladů. Odeslání zprávy je přípustné, je-li uživatel uveden v sekci mynetworks, uživatel je autentizován nebo je adresát zprávy uživatelem domény, která je lokální nebo virtuální v systému, ze kterého je zpráva odesílána. Zjednodušeně se jedná o situaci, kdy je zpráva odesílána v rámci lokálního serveru.

Dalšími možnostmi, kdy je možné nastavit jednotlivá omezení, jsou například smtpd_client_restrictions (Okamžik při spojení), smtpd_helo_restrictions (okamžik při zasílání příkazu HELO/EHLO) a smtpd_sender_restrictions (okamžik při zaslání příkazu MAIL FROM).

U některých poskytovatelů připojení bývá blokován port 25, na kterém funguje ve výchozím stavu SMTP. Pro tento případ bude doplněna následující konfigurace do souboru /etc/postfix/master.cf, která zavede podporu zabezpečeného připojení na portu 465.

```
smtps     inet  n       -       n       -       -       smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o milter_macro_daemon_name=ORIGINATING
```

## 6.5 Instalace Dovecot

Tato podkapitola s instalací Dovecotu je zpracována převážně podle zdrojů [24], [26], [27], [28], [29], [30].

Postfix je v tuto chvíli již nastaven, avšak zbývá ještě nainstalovat a nastavit pro funkční konfiguraci také program Dovecot. Instalace potřebných balíků je provedena zadáním příkazu.

```
aptitude install dovecot-imapd dovecot-pop3d dovecot-common
```

Hlavní konfigurační soubor programu Dovecot /etc/dovecot/dovecot.conf obsahuje mnoho předpřipravených nastavení, u kterých bude stačit pouze provést dílčí změny a doplnění.

První změnou bude nastavení podporovaných protokolů. V tomto případě je požadována podpora protokolů IMAP a POP a to i v zabezpečené verzi za použití SSL.

```
protocols = imap imaps pop pops
```

Aby spojení pomocí těchto protokolů bylo funkční, je nutné ještě odkomentovat a vyplnit část konfigurace, která se věnuje nastavení portů, na kterých protokoly budou naslouchat. Budou použity standartní porty, na kterých probíhá s použitými protokoly komunikace i na ostatních serverech.

```
protocol imap {
     listen = *:143
     ssl_listen = *:993
   }


protocol pop3 {
     listen = *:110
     ssl_listen = *:995
   }
```

Další částí konfigurace je nastavení složky, ve které se nacházejí e-mailové zprávy. Po vzoru předchozího nastavení v Postfixu bude nastaveno stejné umístění.

```
mail_location = maildir:~/Maildir
```

Z důvodu kompatibility s některými e-mailovými klient je ještě nutné povolit autentizaci v prostém textu nastavením příslušného parametru.

```
disable_plaintext_auth = no
```

Aby bylo možné Postfixu poskytnout ze strany Dovecotu podporu pro SASL, je nutné výchozí konfiguraci auth default nahradit upravenou, která bude zajišťovat přihlašování za pomocí uživatelského jména a hesla přímo ze systému.

Celý kód autentizace bude nahrazen následujícím.

```
auth default {
   mechanisms = plain login
   passdb pam {
   }
   userdb passwd {
   }
   socket listen {
      client {
         path = /var/spool/postfix/private/auth
         mode = 0660
         user = postfix
         group = postfix
      }
   }
}
```

Nastavení mechanism označuje povolené mechanismy pro Postfix SMTP. Řádek path nastavuje Dovecot SASL socket a následující řádky omezují přístup pouze pro skupinu a uživatele postfix.

Pro dokončení nastavení autentizace je potřeba dále editovat soubor /etc/default/saslauthd.

```
nano /etc/default/saslauthd
```

V tomto souboru stačí hodnotu START nastavit na yes, aby došlo ke spuštění.

```
START=yes
```

Další změny zajistí možnost komunikace Postfixu se sslauthd. Nejprve je nutné zrušit stávající lokaci SASL. Následně místo ní vytvořit novou s Postfixovým chroot a na závěr nastavit symlink na původní smazanou.

```
rm -r /var/run/saslauthd/
mkdir -p /var/spool/postfix/var/run/saslauthd
ln -s /var/spool/postfix/var/run/saslauthd /var/run
```

Dále následuje změna skupiny pro vytvořenou složku a přidání uživatele postfix do skupiny SASL.

```
chgrp sasl /var/spool/postfix/var/run/saslauthd
adduser postfix sasl
```

Na závěr je potřeba všechny nastavované součásti restartovat, aby došlo k načtení změněných konfigurací. Po restartu je již e-mailový server schopen odesílat a přijímat zprávy.

```
/etc/init.d/dovecot restart
/etc/init.d/postfix restart
/etc/init.d/saslauthd restart
```

## 6.6 Instalace AMaVis, ClamAV a SpamAssassin

Tato podkapitola je zpracována převážně podle zdroje [31].

Pro antivirovou a antispamovou kontrolu v rámci e-mailového serveru bude použita kombinace programů AMaViS, ClamAV a SpamAssassin. Jedná se o velmi častou kombinaci využívanou na e-mailových serverech. Poskytuje velké možnosti nastavení, nevýhodou je však vyšší náročnost na server.

Nejprve je potřeba stáhnout všechny potřebné balíky. Jedná se o velké množství balíků, instalace je proto časově náročnější.

```
apt-get install amavisd-new spamassassin clamav clamav-daemon arj zoo
nomarch cpio lzop cabextract apt-listchanges libnet-ldap-perl libauthen-
sasl-perl  libdbi-perl libmail-dkim-perl p7zip rpm unrar-free libsnmp-
perl
```

Po případy, kdy je již e-mailový server využíván nebo je pravděpodobnost příchodu nějakých zpráv v době provádění změn v konfiguraci, je dobré v Postfixu aktivovat následující volbu.

```
postconf -e soft_bounce=yes
```

Tato volba je tu pro případy, kdy by se Postfix rozhodl odmítnout příchozí e-mailovou zprávu, což v době nastavování parametrů kontroly obsahu hrozí. Díky této volbě nebude žádná zpráva odmítnuta, ale bude po čase proveden další pokus o její doručení. Jedná se o dočasně povolené nastavení, které bude po odladění nastavení kontrolních řešení opět vypnuto.

Nyní tedy po úspěšné instalaci je nutné editovat jeden z konfiguračních souborů Postfixu /etc/postfix/master.cf. Důvodem editace tohoto souboru je fakt, že Postfix ve výchozím nastavení nenaslouchá na portu 10025. Právě na tomto portu však budou vraceny zpět zkontrolované zprávy od AMaViSu, tudíž je nutné doplnit následující dvě služby.

```
smtp-amavis unix -      -       n     -       2  smtp
    -o smtp_data_done_timeout=1200
    -o smtp_send_xforward_command=yes
    -o disable_dns_lookups=yes
    -o max_use=20

127.0.0.1:10025 inet n    -       n       -       -     smtpd
    -o content_filter=
    -o smtpd_delay_reject=no
    -o smtpd_client_restrictions=permit_mynetworks,reject
    -o smtpd_helo_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o smtpd_data_restrictions=reject_unauth_pipelining
    -o smtpd_end_of_data_restrictions=
    -o smtpd_restriction_classes=
    -o mynetworks=127.0.0.0/8
    -o smtpd_error_sleep_time=0
    -o smtpd_soft_error_limit=1001
    -o smtpd_hard_error_limit=1000
    -o smtpd_client_connection_count_limit=0
    -o smtpd_client_connection_rate_limit=0
    -o
receive_override_options=no_header_body_checks,no_unknown_recipient_check
s,no_milters
    -o local_header_rewrite_clients=
```

Aby byly tyto dvě nové služby spuštěny, je nutné Postfix restartovat. Po restartu již obě služby naběhnou.

```
postfix reload
```

Dále je potřeba Postfix nastavit tak, aby používal službu smtp-amavis běžící na portu 10024. Právě nastavením filtru obsahu na tuto službu dojde k předávání zpráv pro kontrolu. Druhý řádek konfigurace zabraňuje dvojímu doručení zprávy uživateli, které by v případě přesměrovávání mohlo nastat.

```
postconf -e content_filter=smtp-amavis:[127.0.0.1]:10024
postconf -e receive_override_options=no_address_mappings
```

Dalším krokem je editace souboru /etc/amavis/conf.d/15-content_filter_mode, ve kterém je nutné odkomentovat následující řádky. Odkomentováním těchto řádků bude zapnuta kontrola obsahu na přítomnost nevyžádané pošty a virů.

```
@bypass_virus_checks_maps = (
   \%bypass_virus_checks, \@bypass_virus_checks_acl,
\$bypass_virus_checks_re);

@bypass_spam_checks_maps = (
   \%bypass_spam_checks, \@bypass_spam_checks_acl,
\$bypass_spam_checks_re);
```

Pro zprovoznění antivirové kontroly za pomocí produktu ClamAV je nutné vytvořit příslušného uživatele zařazeného do skupiny.

```
adduser clamav amavis
```

Následně je nutné editovat další konfigurační soubor.

```
nano /etc/amavis/conf.d/50-user
```

Do tohoto souboru budou přidány konfigurace, na základě kterých bude probíhat kontrola obsahu e-mailových zpráv.

```
$final_spam_destiny=D_PASS;
$sa_tag_level_deflt  = -1000;
$sa_tag2_level_deflt = 5.0;
$sa_kill_level_deflt = 10;
$sa_dsn_cutoff_level = 10;
$mydomain = 'testmail.cz';
@local_domains_acl = ( ".$mydomain" );
```

Pro načtení všech změn v nastavení je nakonec potřebné příslušné části e-mailového systému restartovat.

```
/etc/init.d/clamav-daemon restart
/etc/init.d/amavis restart
```

Pro ověření funkčnosti detekce antispamu je dobré využít ukázkový spam, který lze zaslat do emailové schránky za pomocí následujícího příkazu přímo z programu PuTTy.

```
sendmail michal@testmail.cz </usr/share/doc/spamassassin/examples/sample-
spam.txt
```

Na závěr je potřeba aktualizovat virovou databázi ClamAV na nejnovější verzi za pomocí příkazu.

```
freshclam
```

Pro pravidelnou automatickou aktualizaci je možné do tabulky plánovače CRON přidat následující záznam.

```
0  1  *  *  *  /usr/local/bin/freshclam –quiet
```

Po přidání tohoto řádku bude aktualizace probíhat již automaticky vždy v 1 hodinu ráno bez potřeby interakce uživatele. Toto řešení je s ohledem na potřebnou pravidelnou aktualizaci virové databáze velmi často využívané.

Na závěr po odladění jednotlivých nastavení kontroly obsahu zpráv je nutné vypnout dočasnou volbu neodmítání e-mailových zpráv následujícím příkazem.

```
postconf -e soft_bounce=no
```
[17]

## 6.7 Instalace SquirrelMail

Tato podkapitola s instalací SquirrelMailu je zpracována převážně podle zdrojů [28], [32], [33], [34].

Pro přístup k elektronické poště bez potřeby instalace jakýchkoliv programů pouze za pomocí internetového prohlížeče bude sloužit nastavený emailový klient SquirrelMail.

V případě tohoto webového e-mailového klienta se jedná o velmi propracovanou internetovou stránku. Z tohoto důvodu je nutné před samotnou instalací tohoto nástroje nainstalovat a zprovoznit webový server Apache. Samozřejmě nesmí chybět doplnění o podporu serverového skriptovacího jazyka PHP5.

```
aptitude install apache2
aptitude install libapache2-mod-php5 php5-cli php5-common php5-cgi
```

Po ověření funkčnosti webového serveru přichází čas na stažení balíčku samotného SquirrelMailu a příslušných lokalizačních souborů pro českou jazykovou variantu.

```
apt-get install squirrelmail squirrelmail-decode squirrelmail-locales
```

Pro zprovoznění české jazykové lokalizace je nutné provést některé dílčí změny v konfiguraci. První změnou je přidání záznamu o českém jazyce do souboru locale.gen.

Tato jazykové lokalizace je vytvořena v kódování ISO-8859-2 a je identifikována za pomocí kódu cs_CZ. Bez nutnosti otevírat soubor v editoru je přidání provedeno následujícím příkazem.

```
echo "cs_CZ ISO-8859-2" >> /etc/locale.gen
```

Následně je nutné aktualizovat lokalizace.

```
/usr/sbin/locale-gen
```

V tento okamžik je SquirrelMail doplněný o potřebné části pro využívání české jazykové lokalizace. Pro načtení konfigurace do webového serveru je nutno editovat konfigurační soubor webového serveru Apache.

```
nano /etc/apache2/apache2.conf
```

[17] Zda byla volba opravdu vypnuta, je možné ověřit v konfiguračním souboru /etc/postfix/main.cf.

Do souboru bude doplněno načtení konfigurace, kterou má v sobě obsaženou SquirrelMail.

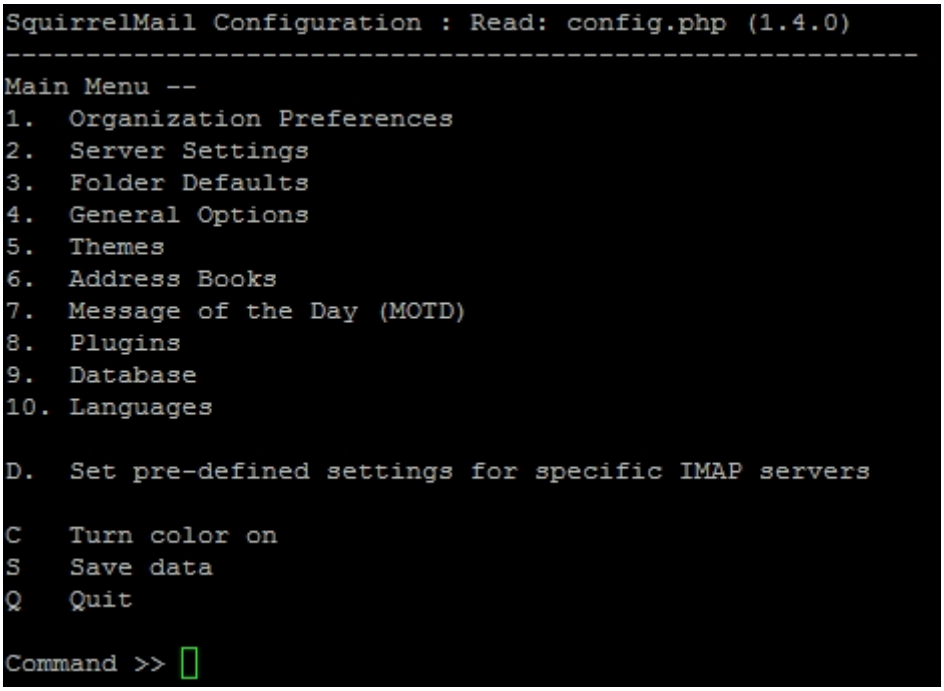
```
Include /etc/squirrelmail/apache.conf
```

Aby se změny projevily, je nutné webový server restartovat. Ten při spuštění načte již i potřebnou konfiguraci.

```
/etc/init.d/apache2 restart
```

Posledním krokem je změna dílčích nastavení samotného SquirrelMailu. Pro změnu je výhodné použít konfigurační utilitu, za pomocí které je změna nastavení velmi jednoduchá a snadno uživatelsky proveditelná. Tato utilita je spuštěna následujícím příkazem.

```
/usr/sbin/squirrelmail-configure
```



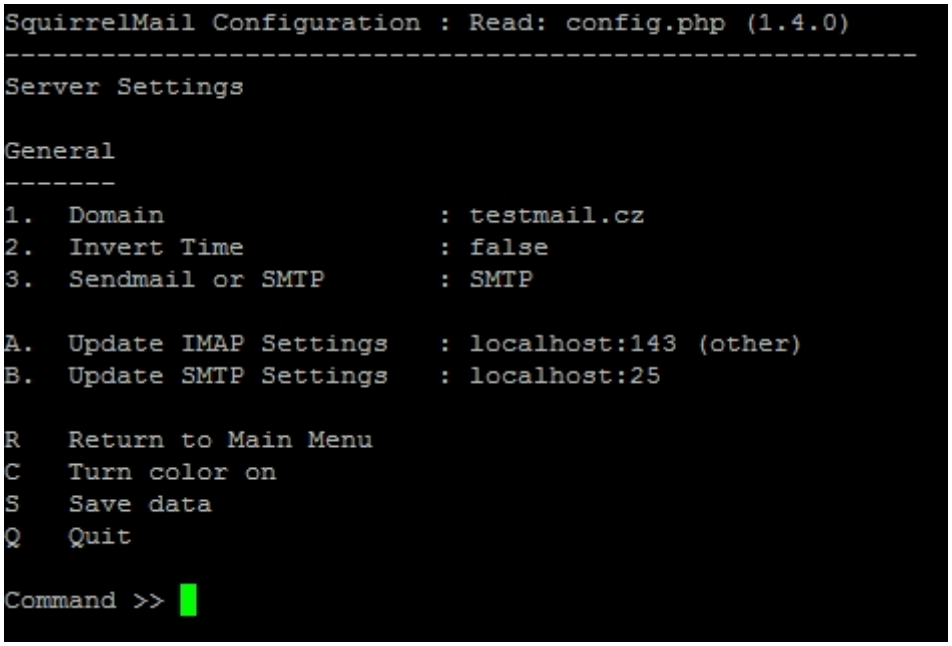

**Obrázek 18 - Úvodní obrazovka konfigurace programu SquirrelMail**

První změna bude provedena v sekci „Server setting“, kde je nutné provést změnu položky „Domain“ na správné nastavení. V tomto případě na název zakoupené domény testmail.cz.

Jelikož všechny součásti e-mailového serveru běží na jednom fyzickém stroji, není nutné měnit nastavení pro IMAP ani SMTP a může tak být použito nastavení localhost.

Z důvodu běhu na jednom stroji taktéž nebude nastavována žádná šifrovaná komunikace mezi e-mailovým serverem a SquirrelMailem. V tomto případě by jednalo o konfiguraci, která by byla zbytečná.

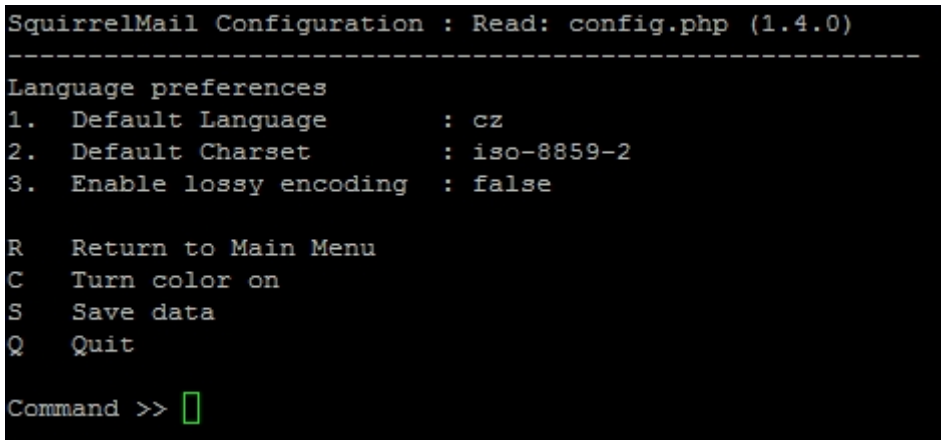
V případě, že by e-mailový server a webový server s tímto online e-mailovým klientem fungovaly na oddělených zařízeních je nutné s ohledem na zabezpečení komunikace tuto volbu nastavit.




**Obrázek 19 - Nastavení serveru programu SquirrelMail**

Další změna v konfiguračním nástroji bude provedena v sekci „Languages“, kde je potřeba nastavit výchozí jazyk na kód cs_CZ a kódování na iso-8859-2. V tomto případě se jedná se pouze o výchozí jazyk nastavený v systému.

V případě potřeby je uživatelsky možné ve webovém rozhraní webmailu provést změnu na libovolnou jazykovou variantu nainstalovanou v systému. Všechna nastavení jednotlivých uživatelů jsou v tomto případě ukládána na serveru.

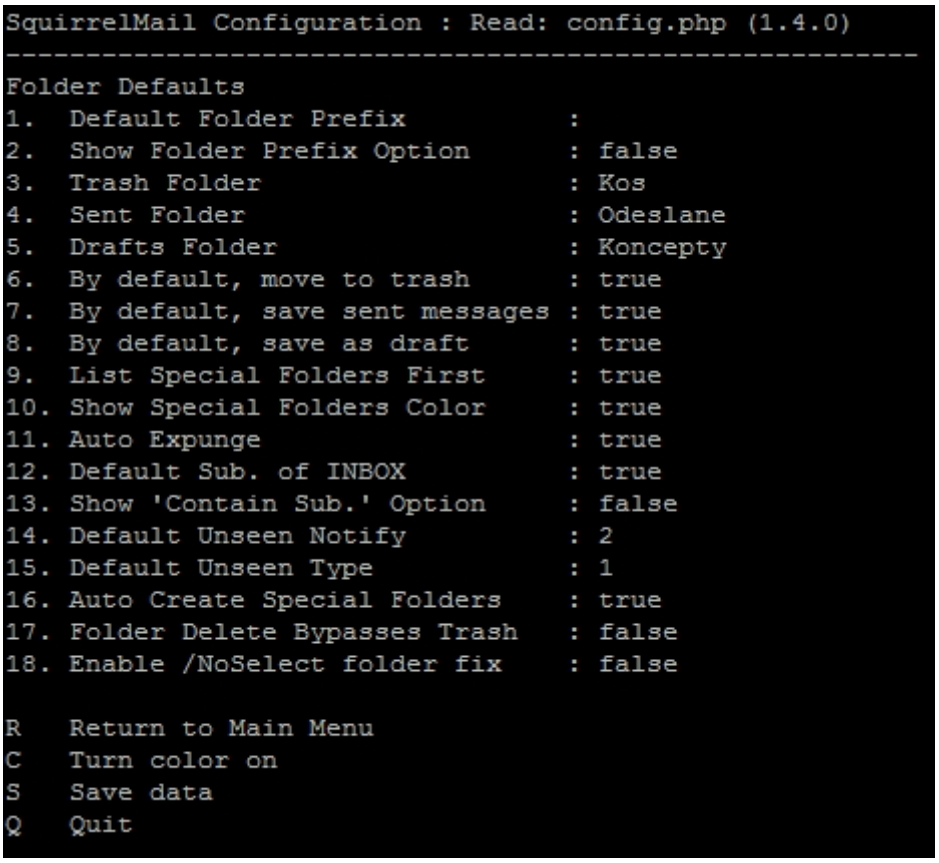



**Obrázek 20 - Nastavení jazykové lokalizace programu SquirrelMail**

Poslední částí editace je změna struktury složek v sekci „Folder defaults“, kdy budou všechny složky přesunuty na vyšší úroveň (nebudou podsložkami složky doručení pošty, kterými jsou ve výchozím stavu).

Ačkoliv by názvy mohly zůstat v anglickém originálu a díky české lokalizaci by byly automaticky překládány, je tato změna zvolena z důvodu využití desktopových klientů (například Microsoft Outlook), kde k přeložení do češtiny nedochází.

Proto budou všechny složky přejmenovány na české varianty (s ohledem na různé e-mailové klienty bez použité diakritiky).

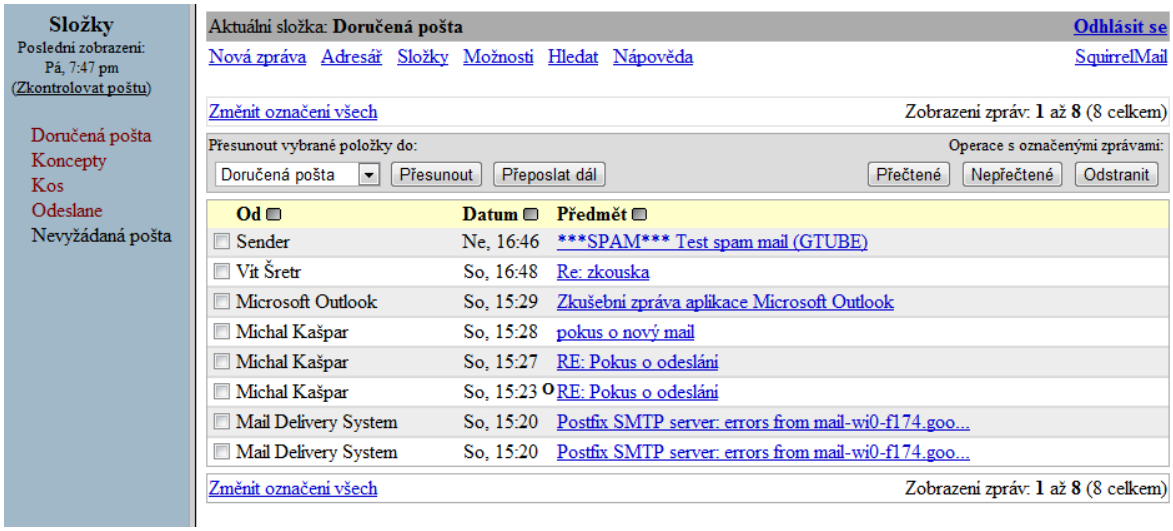



**Obrázek 21 - Nastavení výchozích složek programu SquirrelMail**

Po uložení konfigurace je již webový e-mailový klient funkční. Pro ověření jeho funkčnosti stačí do webového prohlížeče zadat adresu https://testmail.cz/squirrelmail, kde se nachází přihlašovací obrazovka.

Do přihlašovací obrazovky je nutno zadat platné uživatelské jméno a heslo určené pro uživatelský účet v samotném systému Debian, ze kterého jsou uživatelské účty brány.

V případě zadání správných přihlašovacích údajů dojde po stisknutí tlačítka k přihlášení do e-mailové schránky. Po úspěšném přihlášení je zobrazeno webové rozhraní pro práci s elektronickou poštou. Na úvodní straně webmailu je uveden výpis nejnovějších přijatých zpráv.




**Obrázek 22 - Úvodní obrazovka webového klienta**

Jak již bylo názorně ukázáno v kapitole s odposlechem přihlašovacích údajů, tak ve stávající konfiguraci SquirrelMailu je bezpečnostní problém. Ten spočívá v tom, že v aktuálním stavu služba běží na protokolu HTTP. Pro zajištění vyšší bezpečnosti bude proveden převod této služby na HTTPS.

Nejdříve je nutné ověřit, zda je nainstalovaný balík pro ssl-cert. Případně provést jeho doinstalování následujícím příkazem.

```
apt-get install ssl-cert
```

Následně je třeba vybrat umístění, do kterého bude uložen certifikát určený pro zabezpečení přístupu k webu. Pro lepší přehlednost a zapamatovatelnost bude vytvořena složka ssl ve složce webového serveru Apache. Do této složky budou ukládány všechny certifikáty sloužící přístup ke stránkám za pomocí HTTPS.

```
mkdir /etc/apache2/ssl
```

Dalším krokem bude vygenerování certifikátu. Pro veřejné nasazení je vhodnější zakoupení certifikátu od věrohodné certifikační autority, při jehož použití internetový prohlížeč nezobrazuje upozornění.

Pro účely ukázky bude dostačující vygenerování vlastního soukromého certifikátu. Vygenerování se provede zadáním příkazu obsahujícího základní informace o vytvářeném certifikátu. Uložení bude provedeno do předpřipravené složky.

```
openssl req $@ -new -x509 -days 365 -nodes -out
/etc/apache2/ssl/testmailcz.pem -keyout /etc/apache2/ssl/testmailcz.pem
```

Po zadání příkazu ještě následuje zadání dalších informací, které budou použity pro vytvoření certifikátu. Mezi tyto informace patří například země, název organizace a další. Všechny zadávané údaje jsou vidět na následujícím obrázku.

```
root@mail:~# openssl req $@ -new -x509 -days 365 -nodes -out /etc/apache2/ssl/te
stmailcz.pem -keyout /etc/apache2/ssl/testmailcz.pem
Generating a 1024 bit RSA private key
.........++++++
.........++++++
writing new private key to '/etc/apache2/ssl/testmailcz.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CZ
State or Province Name (full name) [Some-State]:Czech republic
Locality Name (eg, city) []:Pardubice
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BP UPCE
Organizational Unit Name (eg, section) []:BP
Common Name (eg, YOUR name) []:testmail.cz
Email Address []:michal@testmail.cz
```

**Obrázek 23 - Proces generování certifikátu**

V tuto chvíli je certifikát vygenerovaný a uložený v požadovaném umístění. Ještě je potřebné nastavit správné oprávnění pro přístup k souboru certifikátu.

```
chmod 600 /etc/apache2/ssl/testmailcz.pem
```

Certifikát je již plně připravený k použití. Dalším krokem je vytvoření virtuálního hosta ve webovém serveru Apache. Konkrétně hosta, který bude naslouchat na portu 443, na kterém funguje zabezpečený protokol HTTP při použití SSL (HTTPS).

Konfigurace virtuálního hosta bude nastavena v souboru /etc/apache2/sites-available/ssl. Pro zápis konfigurace bude opětovně použit textový editor.

```
nano /etc/apache2/sites-available/ssl
```

Obsahem vytvářeného souboru bude následující konfigurace.

```
NameVirtualHost *:443
<VirtualHost *:443>
  ServerAdmin michal@testmail.cz
  ServerName testmail.cz
  ServerAlias *.testmail.cz
  SSLEngine On
  SSLCertificateFile /etc/apache2/ssl/testmailcz.pem

  DocumentRoot /var/www/
  <Directory />
    Options FollowSymLinks
    AllowOverride All
  </Directory>

  <Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
```

```
    Allow from all
  </Directory>

  LogLevel warn
  ErrorLog /var/log/apache2/ssl-error.log
  CustomLog /var/log/apache2/ssl-access.log combined

</VirtualHost>
```

V této konfiguraci je zapnutý SSL engine a nastavený certifikát. Dále jsou uvedeny základní nastavení složky www a cesty pro soubory logů.

Webový server Apache používá pro veřejně dostupné stránky složku sites-enabled, je tedy nutné vytvořit symbolický odkaz pro zajištění funkčnosti vytvářeného virtuálního hosta.

```
ln -s /etc/apache2/sites-available/ssl /etc/apache2/sites-enabled/ssl
```

Dalším krokem je zapnutí podpory SSL a zavedení podpory mod rewrite ve webovém serveru Apache.

```
a2enmod ssl
a2enmod rewrite
```

Mod rewrite slouží pro automatické přesměrování z nezabezpečené verze webových stránek na zabezpečenou. Tímto způsobem budou ošetřeny případy, kdy uživatel zadá adresu na nezabezpečenou verzi webových stránek. Právě pro účely automatického přesměrovávání má v sobě již samotný SquirrelMail předpřipravenou konfiguraci. Stačí editovat soubor /etc/squirrelmail/apache.conf.

```
nano /etc/squirrelmail/apache.conf
```

V konfiguračním souboru je následně nutné vyhledat zakomentovanou část „IfModule mod_rewrite.cz“ a odkomentovat ji.

```
<IfModule mod_rewrite.c>
  <IfModule mod_ssl.c>
    <Location /squirrelmail>
      RewriteEngine on
      RewriteCond %{HTTPS} !^on$ [NC]
      RewriteRule . https://%{HTTP_HOST}%{REQUEST_URI}  [L]
    </Location>
  </IfModule>
</IfModule>
```

Tato část konfigurace v případě zapnutého modulu určeného pro přesměrování provede automatické přesměrování uživatele zabezpečenou verzi webových stránek. Na závěr je potřeba ještě webový server restartovat, aby se projevila všechna nastavení.

```
/etc/init.d/apache2 restart
```

Po restartu webového serveru již automaticky probíhá přesměrování na zabezpečenou verzi webových stránek.

## 6.8 Instalace Postgrey

Tato podkapitola s instalací Postgrey je zpracována převážně podle zdroje [35].

V tomto kroku bude do instalace doplněna podpora o grey listing. Pro tento účel bude použit balíček Postgrey. Nejprve je nutné tento balíček stáhnout a nainstalovat následujícím příkazem.

```
apt-get install postgrey
```

Dalším krokem bude editace konfiguračního souboru balíčku /etc/default/postgrey za pomocí příkazu.

```
nano /etc/default/postgrey
```

V tomto bodě bude doplněna konfigurace, která odesílateli zprávy sdělí, aby e-mail zaslal za 60 sekund znovu.

```
POSTGREY_OPTS="--inet=127.0.0.1:60000 --delay=60"
```

Postgrey je v tento okamžik již nakonfigurovaný, tak je potřeba jej restartovat, aby byla načtena nová konfigurace.

```
/etc/init.d/postgrey restart
```

Nyní je ještě potřeba Postgrey propojit s balíkem Postfix. Protovede se to tak, že do sekce smtpd_recipient_restrictions bude doplněn následující řádek, který zajistí spolupráci s balíčkem Postgrey.

```
check_policy_service inet:127.0.0.1:60000
```

Na závěr je nutné načíst znovu změněnou kongifuraci Postfixu.

```
/etc/init.d/postfix reload
```

V tento okamžik již budou vždy první doručované zprávy z dané adresy odmítány. Po jejich opětovném doručení budou již přijaty a e-mailová adresa bude na dobu 35 dní zařazena do white listu.

## 6.9 Přístup pomocí desktopového klienta

E-mailový server je samozřejmě nakonfigurován nejen pro přístup ke zprávám za pomocí webového SquirrelMailu. Je nastaven pro umožnění přístupu ke schránce za pomocí poštovních protokolů POP, IMAP a SMTP / ESMTP.

Tento způsob připojení tak může být využit například pro připojení za pomocí desktopové aplikace, či pro připojení prostřednictvím dalších přenosných zařízení (smartphone a další).

Aby bylo možné nakonfigurovat rozhraní, je nutné znát přehled adres a portů, na kterých tyto protokoly běží. Přehled všech možných způsobů komunikace je uveden v následující tabulce.

**Tabulka 5 - Přehled způsobů komunikace**

| Typ serveru | Adresa serveru | Protokol | Port | Zabezpečení |
|---|---|---|---|---|
| POP | mail.testmail.cz | POP | 110 | žádné |
| | | | 995 | SSL |
| IMAP | mail.testmail.cz | IMAP | 143 | žádné |
| | | | 993 | SSL |
| SMTP | mail.testmail.cz | SMTP | 25 | žádné |
| | | | 25 | TLS |
| | | | 465 | SSL |

Jak je v tabulce uvedeno, server nabízí pro každý protokol kombinaci nezabezpečeného a zabezpečeného typu spojení na příslušném portu. S ohledem na možnosti odposlechu komunikace je však jednoznačně doporučovanější variantou používání zabezpečených typů spojení.

# Závěr

V rámci této práce bylo poukázáno, že elektronická pošta je velmi používaným a oblíbeným způsobem komunikace i v dnešní době, kdy jí konkurují další způsoby elektronické komunikace. Na pozadí této uživatelsky velmi snadno ovladatelné služby stojí různé technologie, právě proto se jim věnovala úvodní část práce. Byly uvedeny hlavní síťové protokoly a technologie, bez kterých by elektronická pošta patrně nemohla existovat. V jednotlivých částech byly uvedeny nejen základní informace o jejich historii, ale i principy, na kterých pracují, a praktické ukázky jejich využití.

Velmi významná část práce byla zaměřena na potřeby zajištění bezpečnosti využívání této internetové služby. Přestože používání elektronické pošty je velmi jednoduché i pro laika, je nutné dbát na určité zásady bezpečnosti a v rámci zachování bezpečnosti je využívat. Například uvedené používání šifrované komunikace při jakékoliv práci s e-mailovou schránkou a slepé nedůvěřování autentičnosti e-mailové zprávy je základem bezpečnosti. Proto byla zavedena možnost zakoupení elektronického digitálního podpisu, který tyto problémy částečně řeší, a je možné jej využívat při komunikaci s úřady.

Taktéž nesmí být opomenut fenomén dnešní doby, který neustále zaplavuje e-mailovou schránku nejednoho uživatele. Tímto fenoménem je nevyžádaná pošta, se kterou je možné bojovat za pomocí různých antispamových technologií a filtrů. Ať již se jedná o základní ověřování odesílatele zprávy nebo složité algoritmy kontroly samotného obsahu, tak i o velmi důležité technologie neustále vyvíjené se snahou o větší pohodlí uživatele e-mailových služeb. Tyto technologie neustále bojují s inovacemi od lidí rozesílajících tyto nevyžádané zprávy. Velmi značný podíl nevyžádané pošty v rámci e-mailových zpráv v roce 2012 byl poukázán i v provedeném průzkumu.

Na závěr bylo prakticky ukázáno, jakým způsobem zprovoznit vlastní základní e-mailové řešení pro využívání služeb elektronické pošty. Nechybělo zde ani zasazení do reálného prostředí díky virtuálnímu serveru a použité české doméně. Byl brán ohled i na zabezpečení, a tak byla nastavena šifrovaná komunikace nejen v rámci samotných protokolů využívaných například desktopovými klienty, ale i zabezpečení webového e-mailového klienta. Poslední částí konfigurace bylo doplnění antispamového a antivirového řešení, která mají za cíl zajistit ochranu uživatele před viry a odstranit ze schránky podíl nevyžádané pošty.

## Literatura

[1] 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.

[2] BRANDEJS, Michal. Mail. *Materiály k výuce* [online]. © 2008 [cit. 2012-04-21]. Dostupné z: http://www.fi.muni.cz/usr/brandejs/P005/mail.html

[3] DENT, Kyle D. *Postfix: kompletní průvodce*. 1. vyd. Praha: Grada, 2005, 237 s. ISBN 80-247-1029-3.

[4] Reading full email headers. *Gmail Help* [online]. 22. 12. 2011 [cit. 2012-04-21]. Dostupné z: http://support.google.com/mail/bin/answer.py?hl=en&answer=29436

[5] GAŠPAROVIČ, Peter. Elektronická pošta v TCP/IP [8] - MIME. *LINUXZONE* [online]. 13. 06. 2005[cit. 2012-04-21]. Dostupné z: http://www.linuxzone.cz/index.phtml?ids=4&idc=1269

[6] DEVEČKA, Marián. Balancovaný Antispamový Systém [online]. Bratislava, 2009 [cit. 2012-04-22]. Dostupné z: http://www.dcs.fmph.uniba.sk/diplomovky/obhajene/getfile.php/Diplomovka.pdf?id=246&fid=432&type=application%2Fpdf. Diplomová práce. Univerzita Komenského, Fakulta Matematiky, Fyziky a Informatiky.

[7] GAŠPAROVIČ, Peter. Elektronická pošta v TCP/IP [11] - Komunikačné protokoly elektronickej pošty II. *LINUXZONE* [online]. 20. 07. 2005[cit. 2012-04-21]. Dostupné z: http://www.linuxzone.cz/index.phtml?ids=4&idc=1301

[8] GAŠPAROVIČ, Peter. Elektronická pošta v TCP/IP [12] - Komunikačné protokoly elektronickej pošty III. *LINUXZONE* [online]. 26. 07. 2005[cit. 2012-04-21]. Dostupné z: http://www.linuxzone.cz/index.phtml?ids=4&idc=1302

[9] GAŠPAROVIČ, Peter. Elektronická pošta v TCP/IP [10] - Komunikačné protokoly elektronickej pošty I. *LINUXZONE* [online]. 27. 06. 2005[cit. 2012-04-21]. Dostupné z: http://www.linuxzone.cz/index.phtml?ids=4&idc=1280

[10] KANGAS, Erik. How Does Secure Socket Layer (SSL or TLS) Work?. *LuxSci FYI* [online]. 12. 3. 2009 [cit. 2012-04-25]. Dostupné z: http://luxsci.com/blog/how-does-secure-socket-layer-ssl-or-tls-work.html

[11] Elektronický podpis e-mailů. *Elektronický podpis | snadné zřízení* [online]. © 2010 [cit. 2012-04-22]. Dostupné z: http://www.digitalni-podpis.cz/e-mail

[12] Certifikační autorita - popis služeb. *Certifikační autorita PostSignum* [online]. © 2010 [cit. 2012-04-22]. Dostupné z: http://www.postsignum.cz/certifikacni_autorita__popis_sluzeb.html

[13] ADÁMEK, Martin. *Spam: jak nepřivolávat, nepřijímat a nerozesílat nevyžádanou poštu*. 1. vyd. Praha: Grada, 2009. ISBN 978-80-247-2638-0.

[14] WOLFE, Paul, Mike W ERWIN a Charlie SCOTT. *Antispam: metody, nástroje a utility pro ochranu před spamem*. Vyd. 1. Překlad Ivo Fořt. Brno: Computer Press, 2004, 375 s. ISBN 80-251-0479-6.

[15] KOPTA, Martin. Kauza Tvujdum.cz: ty adresy jsme koupili. *Lupa.cz* [online]. 22. 1. 2003[cit. 2012-04-21]. ISSN 1213-0702. Dostupné z: http://www.lupa.cz/clanky/kauza-tvujdum-cz-ty-adresy-jsme-koupili/

[16] ZEMAN, Mirek. Tvujdum dostal pokutu za rozesílání spamu. *Lupa.cz* [online]. 3. 4. 2003[cit. 2012-04-21]. ISSN 1213-0702. Dostupné z: http://www.lupa.cz/clanky/tvujdum-dostal-pokutu-za-rozesilani-spamu/

[17] TLUSŤÁK, Karel. *Problematika emailové komunikace* [online]. Brno, 2008 [cit. 2012-04-22]. Dostupné z: http://www.fit.vutbr.cz/study/DP/rpfile.php?id=6735. Bakalářská práce. Vysokého učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Mgr. Kamil Malinka.

[18] KORTUS, Jaroslav. *Detekce spamu* [online]. Brno, 2006 [cit. 2012-04-22]. Dostupné z: http://is.muni.cz/th/51525/fi_m/diplomka-final.pdf. Diplomová práce. Masarykova univerzita. Fakulta informatiky.

[19] SATRAPA, Pavel. Greylisting: nová metoda boje proti spamu. *Lupa.cz* [online]. 23. 4. 2004[cit. 2012-04-21]. ISSN 1213-0702. Dostupné z: http://www.lupa.cz/clanky/greylisting-nova-metoda-boje-proti-spamu/

[20] KRČMÁŘ, Petr. Greylisting aneb kladivo na spam. *Root.cz* [online]. 3. 8. 2006[cit. 2012-04-22]. ISSN 1212-8309. Dostupné z: http://www.root.cz/clanky/greylisting-aneb-kladivo-na-spam/

[21] KÁRA, Michal. Jak funguje bayesovský antispamový filtr? (1.). *Lupa.cz* [online]. 24. 2. 2005[cit. 2012-04-21]. ISSN 1213-0702. Dostupné z: http://www.lupa.cz/clanky/jak-funguje-bayesovsky-antispamovy-filtr-1/

[22] Big picture. *Workaround.org* [online]. © 2000-2011 [cit. 2012-04-22]. Dostupné z: http://workaround.org/ispmail/squeeze/big-picture

[23] Postfix. *Debian Wiki* [online]. 3. 8. 2011 [cit. 2012-04-22]. Dostupné z: http://wiki.debian.org/Postfix#Installing_and_Configuring_Postfix_on_Debian

[24] Postfix SASL Howto. *The Postfix Home Page* [online]. [2012] [cit. 2012-04-22]. Dostupné z: http://www.postfix.org/SASL_README.html

[25] Postfix Configuration Parameters. *The Postfix Home Page* [online]. [2012] [cit. 2012-04-22]. Dostupné z: http://www.postfix.org/postconf.5.html

[26] Postfix + Dovecot (IMAP/IMAPS) + SASL + Maildir on Debian 6/Ubuntu. *Syslog* [online]. 15. 9. 2011 [cit. 2012-04-22]. Dostupné z: http://syslog.tv/2011/09/15/postfix-dovecot-imapimaps-sasl-maildir/

[27] BURDA, Zdeněk. Mailserver – Postfix, Dovecot a MySQL. *BCVlog* [online]. 11. 9. 2011 [cit. 2012-04-22]. Dostupné z: http://blog.bcvsolutions.eu/mailserver-postfix-dovecot-a-mysql/

[28] Debian Mail Server Setup with Postfix + Dovecot + SASL + Squirrel Mail. *Debian Admin* [online]. 14. 2. 2011 [cit. 2012-04-22]. Dostupné z: http://www.debianadmin.com/debian-mail-server-setup-with-postfix-dovecot-sasl-squirrel-mail.html

[29] Authenticated SMTP. *Workaround.org* [online]. © 2000-2011 [cit. 2012-04-22]. Dostupné z: http://workaround.org/ispmail/squeeze/postfix-smtp-auth

[30] Setting up Dovecot. *Workaround.org* [online]. © 2000-2011 [cit. 2012-04-22]. Dostupné z: http://workaround.org/ispmail/squeeze/setting-up-dovecot

[31] Optional: Content scanning with AMaViS. *Workaround.org* [online]. © 2000-2011 [cit. 2012-04-22]. Dostupné z: http://workaround.org/ispmail/squeeze/content-scanning-amavis

[32] ŠAUR, Jindra. Squirrelmail a čeština. *Dino.Saur.Cz* [online]. 18. 8. 2008 [cit. 2012-04-22]. Dostupné z: http://dino.saur.cz/squirrelmail-a-cestina

[33] Debian, Apache2 + ssl + mod_rewrite. *Jens.cz* [online]. 26. 2. 2008 [cit. 2012-04-22]. Dostupné z: http://www.jens.cz/debian-apache2-ssl-mod_rewrite/

[34] Install and Configure Apache2 with PHP5 and SSL Support in Debian Etch. *Debian Admin* [online]. 18. 12. 2008 [cit. 2012-04-22]. Dostupné z: http://www.debianadmin.com/install-and-configure-apache2-with-php5-and-ssl-support-in-debian-etch.html

[35] Killing That Spam With Postgrey And Postfix. *HowtoForge - Linux Howtos and Tutorials* [online]. 28. 6. 2006 [cit. 2012-04-27]. Dostupné z: http://www.howtoforge.com/greylisting_postfix_postgrey

## Příloha a – Zdrojový kód souboru index.php

```
<?php

function __autoload($class) {
    if (file_exists('class/' . $class . '.php')) {
        require_once ('class/' . $class . '.php');
    }
}

if (isset($_POST["odesilatelJmeno"])) {
    if ($_POST["odesilatelJmeno"] != "" && $_POST["odesilatelAdresa"] !=
"" && $_POST["prijemceAdresa"] != "" && $_POST["predmetEmailu"] != "" &&
$_POST["textEmailu"] != "") {
        $email = new Email($_POST["odesilatelJmeno"],
$_POST["odesilatelAdresa"], $_POST["prijemceAdresa"],
$_POST["predmetEmailu"], $_POST["textEmailu"]);

        $email->send();
    }
}
?>

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8" />
        <link rel="stylesheet" type="text/css" href="css/style.css" />
        <title>Podvržení e-mailu za pomoci PHP</title>
    </head>
    <body>
        <form action="index.php" method="POST">
            <ul>
                <li><label>Jméno odesílatele</label><input type="text"
name="odesilatelJmeno"/></li>
                <li><label>Adresa odesílatele</label><input type="text"
name="odesilatelAdresa"/></li>
                <li><label>Adresa příjemce</label><input type="text"
name="prijemceAdresa"/></li>
                <li><label>Předmět e-mailu</label><input type="text"
name="predmetEmailu"/></li>
                <li><label>Text e-mailu</label><textarea
name="textEmailu"></textarea></li>
                <li><input type="submit" value="Odeslat"/></li>
            </ul>
        </form>
    </body>
</html>
```

## Příloha B – Zdrojový kód souboru Email.php

```
<?php

class Email {

    private $jmenoOdesilatele = null;
    private $adresaOdesilatele = null;
    private $adresaPrijemce = null;
    private $predmetEmailu = null;
    private $textEmailu = null;

    public function __construct($jmenoOdesilatele, $adresaOdesilatele,
$adresaPrijemce, $predmetEmailu, $textEmailu) {
        $this->jmenoOdesilatele = $jmenoOdesilatele;
        $this->adresaOdesilatele = $adresaOdesilatele;
        $this->adresaPrijemce = $adresaPrijemce;
        $this->predmetEmailu = $predmetEmailu;
        $this->textEmailu = $textEmailu;
    }

    private function codeToUtf8($text) {
        return "=?utf-8?B?" . base64_encode($text) . "?=";
    }

    public function send() {
        if ($this->jmenoOdesilatele != null && $this->adresaOdesilatele
!= null && $this->adresaPrijemce != null && $this->predmetEmailu != null
&& $this->textEmailu != null) {
            $this->predmetEmailu = $this->codeToUtf8($this-
>predmetEmailu);
            $this->jmenoOdesilatele = $this->codeToUtf8($this-
>jmenoOdesilatele);

            $header = "MIME-Version: 1.0\n";
            $header .= "Content-Transfer-Encoding: QUOTED-PRINTABLE\n";
            $header .= "Content-type: text/html; charset=utf-8\n";
            $header .= "From: \"" . $this->jmenoOdesilatele . "\" <" .
$this->adresaOdesilatele . ">\n";
            if (mail($this->adresaPrijemce, $this->predmetEmailu, $this-
>textEmailu, $header)) {
                return true;
            } else {
                return false;
            };
        }
    }
}

?>
```

## Příloha C – Zdrojový kód souboru style.css

```
body {
    font-family: Arial, sans-serif;
    font-size: 13px;
}

form fieldset{
    border:none;
}

form ul{
    list-style:none;
}

form ul li{
    padding: 15px 0 15px 0;
    clear:both;
}

form label{
    display: block;
    padding:0 10px 0 0;
    float:left;
    width:150px;
    border-bottom:  1px solid #CBD1D8;
    color:#333333;
}

form input[type="text"]{
    float: left;
    width:400px;
    height:25px;
    border:  1px solid #CBD1D8;
    padding: 0 10px 0 10px;
    color:#333333;
}

form input[type="submit"]{
    padding: 10px 20px 10px 20px;
    margin: 10px 0 0 490px;
    border:  1px solid #CBD1D8;
    background-color: #bbc0c7;
    float:left;
    cursor:pointer;
    color:#333333;
}

form textarea{
    float: left;
    width:400px;
    height:300px;
    border:  1px solid #CBD1D8;
    padding: 10px 10px 10px 10px;
    color:#333333;
}
```