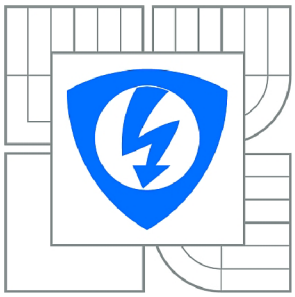
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

ANTISPAMOVÉ FILTRY

ANTISPAM FILTERS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

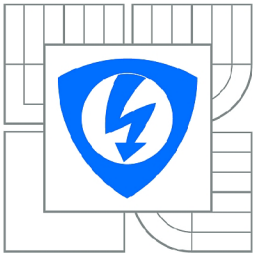
AUTOR PRÁCE — Bc. JIŘÍ FRANTIŠEK

AUTHOR

VEDOUCÍ PRÁCE — Ing. RADIM PUST

SUPERVISOR

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jiří František ***ID:*** 74879
Ročník: 2 ***Akademický rok:*** 2010/2011

NÁZEV TÉMATU:

Antispamové filtry

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je popsat princip činnosti současně dostupných modulů antispamového filtru. Dále navrhnout a realizovat řešení antispamového filtru pro poštovní server na platformě operačního systému Linux. V rámci realizace řešení by student měl vytvořit webové rozhraní pro správu a konfiguraci antispamového filtru.

DOPORUČENÁ LITERATURA:

[1] WOLFE, Paul, SCOTT, Charlie, W. ERWIN, Mike. Antispam : Metody, nástroje a utility pro ochranu před spamem. [s.l.] : [s.n.],
2005. 376 s. ISBN 80-251-0479-6.
[2] HILDEBRANDT, Ralf, KOETTER, Patrick. Postfix : Provozujeme poštovní server v Linuxu. [s.l.] : [s.n.], 2006. 432 s. ISBN
80-251-1020-6.

Termín zadání: 7.2.2011 ***Termín odevzdání:*** 26.5.2011

Vedoucí práce: Ing. Radim Pust

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

UPOZORNĚNÍ:

ABSTRAKT

Práce se zabývá návrhem a realizací antispamového řešení pro operační systém GNU/Linux. V úvodní části je popsána teorie přenosu, příjmu poštovních zpráv a problematika nevyžádané pošty. V rámci práce je realizován poštovní server přenosovým agentem Postfix. Amavis byl využit jako antispamové řešení, který tvoří rozhranní mezi Postfixem a programy na kontrolu obsahu zpráv. Ty jsou tvořeny SpamAssassin a ClamAV. Hlavním cílem bylo vytvoření aplikace pro nastavování antispamových filtrů. Výsledkem je možnost nastavování a vytváření filtrů spolu se seznamy černých a bílých odesílatelů. Zprávy uložené v karanténě je možné obnovit. Zprávy uložené v karanténě slouží pro tvorbu statistik nevyžádané pošty.

KLÍČOVÁ SLOVA

Spam, mail, GNU/Linux, Postfix, Amavis, SpamAssassin, ClamAV, server, webová aplikace

ABSTRACT

The thesis is involves the desing antispam solution for operating system GNU/Linux. At the first is going through theory of transport, receive mail message and problematic of spam. The content of thesis is realize mail server with mail transfer agent Postfix. Amavis was used as antispam solution, which make an interface between Postfix and content checkers. This was created by SpamAssassin and ClavAV. Main goal was created aplication for setting of antispam filters. The result is possibility of setting a creating filters with black and white lists. Messages in quarantine can be restored. This messages is used for creating a statistic output from spam mails.

KEYWORDS

Spam, mail, GNU/Linux, Postfix, Amavis, SpamAssassin, ClamAV, server, web aplication

FRANTIŠEK J. *Antispamové filtry*. Místo: Brno. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 69 s. Vedoucí diplomové práce byl Ing.Radim Pust.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Antispamové filtry“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne ....

(podpis autora)

Děkuji vedoucímu práce panu Ing.Radimu Pustovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

Lidé mezi sebou komunikují již od pradávna, dokonce i v době, kdybychom jim lidé pravděpodobně neříkali. V jednotlivých fázích vývoje člověka se jednalo o různé dorozumívací prostředky začínaje posunky, skřeky, malbami až po kouřové signály a lidskou řeč. Každá tato etapa využívala komunikační prostředky aktuální doby. V druhé polovině 20. století odstartovala fáze komunikace lidí skrze počítače.

Protokol SMTP, který byl vytvořen v roce 1982, slouží jako základ pro vyměňování zpráv ve formě e-mailů. Postupným rozšiřovaním sítě se začali objevovat ve schránkách uživatelů zprávy, které ze začátku nikdo nepovažoval za problém. Tyto zprávy neměli pro cílového adresáta žádnou informační hodnotu a s dalším vývojem UBE (Unsolicited Bulk Email) docházelo k plnění e-mailových schránek reklamními UCE (Unsolicited Commercial Email) zprávami.

Spolu se vznikem nevyžádaných e-mailů se začali vytvářet systémy sloužící k omezení množství těchto zpráv ve schránkách uživatelů. Jsou založeny na principech analýzy obsahu a průběhu spojení při odesílání zprávy. Obě tyto metody jsou předmětem řešení diplomové práce.

Práce je rozdělena do pěti kapitol. První dvě kapitoly rozebírají teoretický základ poštovní komunikace, používané protokoly a problematiku nevyžádané pošty. Třetí a čtvrtá kapitola se věnuje tvorbě poštovního serveru s antispamovou ochranou. Poslední kapitola popisuje vytvořenou aplikaci na nastavování antispamových politik.

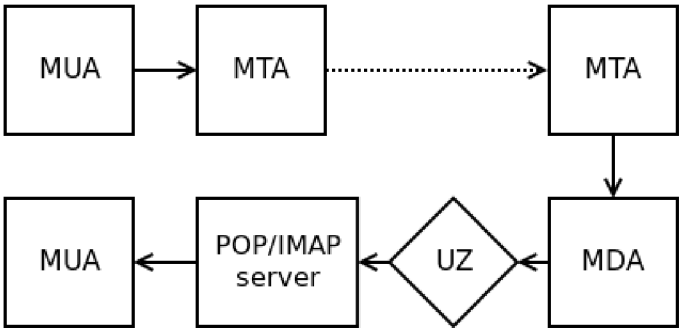
1 TEORETICKÝ ZÁKLAD

1.1 Obecný princip komunikace

Princip vyměňování elektronických zpráv je složen z několika funkčních bloků. Ty mají za úkol spolu komunikovat a předávat informace od odesílatele k příjemci. Celý tento proces je znázorně na obrázku 1.1.

- MUA poštovní uživatelský agent – Message User Agent
- MTA poštovní přenosový agent – Message Transfer Agent
- MDA poštovní doručovací agent – Message Delivery Agent
- Uložiště zpráv

MUA zpracovává požadavky uživatele a převádí je do formy vhodné pro přenos. Hlavním úkolem je vytvořit obousměrný komunikační kanál s MTA, předat mu zprávu a čekat na potvrzení o přijetí. Nejčastěji se jedná o e-mailového klienta např. Evolution Mail, Microsoft Outlook, Thunderbird atd. Moderní programy implementují rozšiřující funkce o kontaktní adresář, kalendář, čtečku RSS kanálů nebo e-mailových diskuzí. Jakmile MTA přijme e-mail, stává se zodpovědným za doručení



Obr. 1.1: Schéma poštovní komunikace

nebo informování o nedoručitelnosti. Jeho starostí je vyhledat cílový MTA, se kterým může komunikovat přímo nebo předá zprávu dalšímu uzlu. Těchto uzlů může být několik a označují se jako Mail Relay neboli předávači zpráv. Mezi nejrozšířenější MTA patří Sendmail, Postfix, Microsoft Exchage, Lotus Domino atd.

Po přijmutí zprávy cílovým agentem MTA je zpráva předána do MDA, který je uloží do lokálního nebo síťového uložiště dat (UZ). Odsud je již zpráva dostupná pro uživatelské MUA pomocí dostupných protokolů [3].

1.2 Používané protokoly

Systém doručování elektronické pošty je značně modulární a může se skládat z mnoha protokolů. Ty mohou být proprietární nebo všeobecně standardizované. Mezi nejčastěji používané patří protokoly definované organizací IETF, která vydává jejich specifikace pomocí tzv. RFC (Request for comments).

1.2.1 SMTP

Úkolem protokolu SMTP je poslat datovou zprávu efektivně a spolehlivě. Toho je dosáhnuto transportním protokolem TCP, který je spolehlivý co se týče přenosu dat a má navíc mechanismy pro řízení kontroly toku. Pomocí protokolu SMTP komunikují jednotlivá MTA mezi sebou nebo MUA s MTA.

SMTP příkazy

Jakmile je vytvořen komunikační kanál začne klient navazovat komunikaci se serverem. Ta je tvořena textovými příkazy, které si mezi sebou vyměňují. Princip komunikace je založen na dotazu a odpovědi, kdy jednotlivé dotazy musí čekat na odpověď a až potom je možné položit další. Mezi základní dotazy patří [17]:

- `EHLO` - Jedná se o rozšíření příkazu `HELO` používaného ve starších specifikací protokolu SMTP. Slouží k ověření funkčnosti e-mailového serveru na protilehlé straně. Argumentem příkazu je plně kvalifikované doménové jméno cílového systému. Pokud je vše v pořádku, bude vrácena hodnota `250 OK`.
- `MAIL` - Udává tzv. zpětnou cestu k adresátovi neboli adresu od kterého odesílatele e-mail přišel. Na tuto adresu se posílají odpovědi nebo informace o nedoručitelnosti e-mailu (bounce zprávy). Ve specifických případech se nepoužívá z důvodu možného vytvoření smyček.
- `RCPT` - Určuje příjemce e-mailu. Pokud je jich více, příkaz `RCPT` je použit více krát. Dříve bylo využíváno pro označení cesty k příjemci.
- `DATA` - Samotný obsah zprávy ukončený znaky `<CRLF>.<CRLF>`
- `QUIT` - Ukončuje SMTP relaci ze strany odesílatele. Přijímací strana musí počkat než dostane příkaz `QUIT` na který odpoví hodnotou `250 OK` a tím se považuje spojení za ukončené.

Na každý dotaz musí být přesně jedna odpověď. Jedná se o trojmístné číselné hodnoty následované textovou informací, která je pouze pro účely snadnějšího rozpoznávání chyb a není nijak standardizována (až na některé výjimky hlavních chybových kódů). Základní třídy kódů jsou:

- 1XX Předběžná pozitivní odpověď: Příkaz byl přijat, ale čeká se na potvrzení údajů. Klient by měl poslat další příkaz a uvést jestli pokračovat nebo ukončit dotaz.
- 2XX Pozitivní odpověď: Požadovaná akce byla dokončena.
- 3XX Částečná pozitivní odpověď: Příkaz byl přijat, ale požadovaná akce je pozdrženo až do obdržení dalších informací. Nejčastěji používané v dotazu `DATA`
- 4XX Dočasná chyba: Žádost nebyla přijata a nebylo ji možné provést. Tato chyba může být dočasná a po opakování celého procesu může skončit úspěšně.
- 5XX Trvalá chyba: Žádost nebyla přijata a nebylo ji možné provést. Jedná se o trvalou chybu a její náprava je možná až po změně předchozích kroků.

Příklad SMTP komunikace

Pro názorný příklad použití příkazů použijeme připojení na poštovní server pomocí programu Telnet a jako cílový port zvolíme 25, na kterém přijímá server příchozí spojení. Doménové jméno serveru je titan.jirifrantisek.cz.

```
telnet titan.jirifrantisek.cz 25
Server: 220 titan.jirifrantisek.cz ESMTP Postfix
Klient: EHLO titan.jirifrantisek.cz
Server: 250 titan.jirifrantisek.cz
Klient: MAIL FROM:<jirka@jirifrantisek.cz>
Server: 250 OK
Klient: RCPT TO:<info@jirifrantisek.cz>
Server: 250 OK
Klient: DATA
Server: 354 END DATA WITH <CRLF>.<CRLF>
...
Klient: .
Server: 250 OK
Klient: QUIT
SERVER: 221 GOODBYE
```

Popis komunikace je uveden v kapitole o Postfixu.

1.2.2 DNS

Na začátku fungování protokolu SMTP vypadaly celosvětové sítě úplně jinak než dnes. Servery se připojovali pomocí vytáčených linek, mnohdy pouze přes večer, kdy

bylo připojení levnější. Navíc do některých částí sítě vedla pouze jediná cesta. To vedlo k dlouhé době přenosu zprávy a musela se dokonce pomocí příkazu „MAIL“ specifikovat cesta přepravními uzly k cílovému adresátovi [27].

V současné době si již nemusíme pamatovat cestu, kterou musí podstopit e-mailová zpráva ke svému příjemci. Pro směrování e-mailů používáme službu DNS (Domain Name System), jejíž schopností je zjistit IP adresu cílového systému z doménového jména. DNS využívá několik typů záznamů, pro účely přenosu elektronické pošty se jedná o tyto [22]:

- A záznam - prosté přeložení doménového jména na IP adresu
- Kanonický záznam - převádí jedno doménové jméno na druhé
- MX záznam - uvádí poštovní servery požadované domény
- PTR záznam - slouží pro reverzní překlad z IP adresy na doménové jméno

Funkci DNS je možné vyzkoušet programem nslookup. Na systému Linux jej spustíme stejnojmenným příkazem nslookup, tím se dostaneme do interaktivního režimu. Pomocí příkazu set type=mx vybereme dotaz na IP adresu poštovního serveru a pak zadáme cílovou doménu. Výstup může vypadat např. takto:

```
jirifrantisek.cz mail exchanger = 10 titan.jirifrantisek.cz.
```

Pro doménu jirifrantisek.cz existuje jeden e-mailový server s doménovým jménem titan.jirifrantisek.cz a jeho priorita je 10. Pokud by neexistoval žádný mx záznam, systém nám vrátí hodnotu A záznamu a měli bychom zkusit poslat zprávu na něj. Pokud existuje alespoň jeden mx záznam, je opatřen hodnotou priorita, která udává preferenci daného serveru. Nižší hodnota má přednost. Jestli doména obsahuje více záznamů se stejnou prioritou, úkolem odesílatele je vybrat náhodně jeden ze serverů (např. metodou Round-robin). Následným dotazem na záznam A, zjistíme že adresa serveru titan.jirifrantisek.cz je:

```
Name: titan.jirifrantisek.cz
Address: 80.188.107.218
```

Tím je proces hledání cílového serveru u konce.

1.2.3 POP

Jedná se o protokol, kterým se stahuje došlá pošta ze serveru do poštovního klienta. Existuje v několika verzích, nejrozšířenější je třetí. Standardně POP server naslouchá na portu číslo 110, pokud chceme vlastníky schránek autorizovat pomocí TLS nebo SSL může být komunikace směrována na port 995 [23].

Jednoduchost POP spočívá v tom, že hlavním cílem je otevřít schránku, stáhnout všechny došlé zprávy do klienta, smazat je ze serveru a relaci ukončit. V poslední době má mnoho poštovních klientů možnost zanechat zprávy na serveru. Takový způsob práce s poštou má výhodu v rychlém přesunu všech zpráv s možností se brzy odpojit a zpracovat poštu nezávisle na stavu připojení k síti.

Práce s poštovní schránkou pomocí protokolu POP se skládá z několika fází:

- Připojení - Navázání komunikace se serverem.
- Autorizace - Ověření uživatele pro přístup do schránky.
- Transakce - Samotná práce se schránkou - stažení a smazání došlých zpráv.
- Aktualizace - Provedení úkolů definovaných v předchozí části a ukončení relace.

1.2.4 IMAP

Podobně jako protokol POP slouží IMAP k přístupu k e-mailové schránce. Nejrozšířenější verzí je IMAP v4 první revize. IMAP server poslouchá na portu 143, při komunikaci přes TLS nebo SSL se jedná o port 993.

Na rozdíl od POP je přístup do schránky přes IMAP více interaktivní. Je možné nastavit stahování pouze záhlaví nebo celých zpráv. Hlavní výhodou je využití při přístupu více klientů k jedné schránce. Jelikož zprávy zůstávají na serveru tak s nimi můžeme pracovat z více počítačů (např. z práce a z domu). E-maily obsahují značky, zda-li byla zpráva již přečtená, smazaná atd. Tyto značky se synchronizují mezi poštovními klienty. Schránka může obsahovat více složek pro zpřehlednění doručené pošty. IMAP dokonce obsahuje rozhraní pro možnost přidání rozšíření protokolu [15].

1.2.5 Možnosti autorizace

Autorizovat uživatele pro přístup k funkcím je možné pomocí několika metod [3]:

Prostý text

Jedná se o nejjednodušší metodu, kdy se přihlašovací údaje posílejí v prostém (plain) textu. Největší nevýhodou je snadná možnost odposlechu přihlašovacích údajů. Stačí mít přístup k části sítě kudy proudí data a vhodné softwerové vybavení.

TLS

TLS (Transport Layer Security) je nástupce protokolu SSL (Secure Sockets Layer). Zajišťuje šifrování komunikace mezi klientem a serverem, integritu dat a důvěryhodnost komunikujících uzlů [4].

2 CO JE TO NEVYŽÁDANÁ POŠTA

Někdy ji nazýváme také anglickým slovem spam. Jedná se o elektronickou poštu, která příjemce takové zprávy obtěžuje a nepřeje si aby ji dostával. Nejčastěji se jedná o reklamní sdělení rozeslané na velké množství adresátů. Slovem spam v současné době nepojmenováváme pouze elektronické dopisy, ale také nevhodné příspěvky na diskuzních fórech, reklamní SMS zprávy atd. Obsahem spamu jsou nejčastěji nabídky na výhodné ceny léků, nejrůznější finanční produkty nebo turistické zájezdy. Většina zpráv je psaná v angličtině. Forma bývá také různá, od prostého textu přes html zprávy až po obrázkové nebo pdf zpracování. Často takové zprávy obsahují virovou infekci [11].

Slušnější formu spamu je možné odhlásit, jenže tato možnost není samospásná. V určitých případech to může pomoci, ale také upozornit, že vaše e-mailová schránka je aktivní a spameři mohou začít posílat větší množství spamu na vaši adresu.

Mohli bychom si položit otázku, proč se spam rozesílá? Jako skoro za každou věcí, tak i za spamem stojí peníze. Sice úspěšnost zpráv není vysoká, ale v množství pošty, kterou vyprodukují spamovací systémy je možné vydělávat. Příjem financí je tvořen z několika druhů výdělku:

- Upozornění na skutečné produkty - Úplně původní druh nevyžádaných zpráv. Slouží pouze k upozornění na danou věc s možností její koupě.

- Příjem z reklamy - Cílem zprávy není cokoliv prodávat, nýbrž pouze nalákat co nejvíce uživatelů internetu na webové stránky plné reklamy, z které plynou peníze za množství jejího shlédnutí. Další možností je nakažení napadnutých stanic při přístupu na danou stránku viz. kapitola 2.3.1.

- Tvorba adresních databází - Prvním úkolem tvůrce nevyžádané pošty je vytvořit databázi kontaktů, která se pak využívá pro posílání spamu. Takovou databázi je pak možné prodat pro další využití.

- Podvody, tzv. phishing - Jedná se o zamaskování zprávy takovým způsobem, že vypadá jako by pocházeli např. z banky. Obsahuje odkaz do falešného internetového bankovnictví a pokud zde použijete platné přihlašovací údaje tak budou použit pro převod peněz z vašeho skutečného účtu.

2.1 Jaké techniky využívají spameři

2.1.1 Vytváření databází

Základní stavební kámen nevyžádané pošty tvoří samozřejmě elektronická adresa. Pokud nechceme, aby jsme měli poštovní schránku plnou spamu, měli bysme být s jejím zveřejňováním co nejvíce opatrní [11].

Nejjednodušší způsob jak získat e-mailové adresy je prohledávat pomocí programů, tzv. spambotů webové stránky a extrahovat z nich údaje připomínající tvar elektronické adresy. Měli bychom být proto opatrní kam píšeme naši adresu. Pokud je to nevyhnutelné, měli bychom se aspoň snažit ji zveřejňovat v trošku rozdílné formě. Nejčastěji se zaměňuje znak „@“ nebo „.“ za přepis jeho skutečného významu. Taková e-mailová adresa pak vypadá:
`jirka (zavináč) jirifrantisek (tečka) cz`
Existuje mnoho dalších způsobá jak zaměnit tyto charakteristické znaky e-mailové adresy. Někdy je adresa zobrazována také jako obrázek.

Dalším způsobem získávání adresy je hrubý útok. Pokud některé domény obsahují mnoho adres (např. freemailové služby) tak se používá hrubý útok, kdy se zkouší všechny možné kombinace písmen k sestavení adresy. Sofistikovanějším řešením může být slovníkový útok, kdy je použita databáze pravděpodobných slov, které by mohli tvořit adresu.

Údržba databáze

Po určitém čase, kdy útočník získává více a více adres, je potřeba jeho databází také určitým způsobem spravovat. K aktuálnosti databáze se používají techniky ověření aktuálnosti adresy [11]:

- Potvrzení o přečtení - Vetšina moderních e-mailových klientů umožňuje funkci ověření o přečtení. Pokud je ověření odesíláno automaticky, můžeme útočníkovi sdělovat informaci o používání schránky.
- Odhlašovací formulář - Jak už bylo zmiňováno dříve, část spamových zpráv obsahuje možnost odhlášení odběru spamové zprávy. Pokud ji využijeme, riskujeme zvýšení množství spamu v naší schránce.

2.1.2 Tématické zprávy

Spam je tvořen inzercí na léky až z 74% [9], mnoho lidí již takové nabídky nepotřebuje nebo je rovnou odsuzuje. Změnu v zaměření spamu můžeme pozorovat při velkých celosvětových akcích. Aktuálně se jedná o světový šampionát ve fotbalu nebo politických či přírodních neštěstích (nepokoje v Egyptě, tsunami v Japonsku).

2.1.3 Jak se rozesílá spam

Skrytí identity odesílatele

Pokud chce rozesílatel nevyžádáné pošty svoji činnost provádět delší dobu, musí se krýt nejrůznějšími způsoby, aby se nedal vystopovat [1].

- Maskování vlastní IP adresy - Nazývaný také jako IP spoofing. IP adresa útočníka je zfalšována nebo zakódována, aby byla ztížena jeho možná identifikace.
- Využívání otevřených systémů - Některé e-mailové servery nejsou dostatečně zabezpečeny proti zneužití a umožňují odesílat zprávy komukoliv. Takovým systémům říkáme Open Relay.
- Hledání veřejných sítí - V současné době existuje mnoho veřejně dostupných sítí s přístupem do Internetu. Takové sítě můžou být v hotelech, kavárnách nebo nezabezpečené domácí sítě. Útočník se do takové sítě připojí a vychrlí během krátké doby velké množství spamu a zase se odpojí.

Hledání děr v antispamových filtrech

Jedná se o techniky, kdy se do zprávy přidávají informace, které člověk nevidí, ale antispamovou ochranu dokáží ošálit. Můžou to být různé html značky, záměrné přehazování písmen, použití mezer mezi písmeny nebo nahrazování některých znaků jejich číselnými podobami. Používá se i zosobnění zprávy použitím části e-mailové adresy. Dalším případem může být lokalizace textu do jazyka mateřského pro uživatele (odhadnutého z doménového jména) [1].

2.2 Opatření proti spamu

Nevyžádaná pošta je v současné době velice známý pojem a setkává se s ním skoro každý uživatel Internetu. Jedná se o nešvar, který pravděpodobně nepůjde nikdy vymýtit a bude tu s námi vždy. K pomyslnému sto procentnímu vymýcení spamu se můžeme příblížit více či méně. Důležité je, aby rozhodovací systém raději propustil nějakou nevyžádanou poštu než, aby smazal skutečný mail (tzv. falešné pozitivum).

2.2.1 Omezení na straně odesílatelů

Z důvodu řešení příčiny nežli následků, se začali vytvářet opatření před hromadným rozesíláním spamu na straně odesílatelů. To má za úkol co nejvíce ztížit práci spamerům, ale na druhou stranu bohužel i určitým způsobem omezuje regulérní uživatele.

Autorizace SMTP

Z historických důvodů uvedených na straně 14, bývaly využívány servery Open Relay pro doručování pošty. Tehdy byla komunikace postavena na málo uzlech, které mnohdy byly dostupné pouze po omezenou dobu a všechny servery přeposílaly poštu pro všechny účastníky. Dnes je nutné, aby SMTP servery odesílaly poštu jen pro známé uživatele. Toho může být dosaženo specifikací IP adres nebo celých rozsahů odesílajících uživatelů. Problém nastává pokud, některý z klientů daného systému cestuje a využívá velkého množství různých IP adres. Řešením je autentizace uživatele pomocí přihlašovacího jména (většinou e-mailové adresy) a přihlašovacího hesla. Tyto údaje mohou být přenášeny jako běžný text, což může být náchylné na odposlouchání nebo šifrovány pomocí TLS nebo SSL, více na straně 16.

Příklad omezení

Někteří poskytovatelé internetových služeb ISP (Internet Service Provider) zakazují propustnost portu 25 mimo své sítě. Je to z důvodu, aby se IP adresy z jejich rozsahu nedostaly na blacklisty jako rozesílatelé spamu. Místo toho poskytují vlastní SMTP servery, které můžou používat jejich zákazníci. Obvykle je možné tyto servery používat i mimo jejich sítě, v tom případě je nutná autorizace zákazníka. Komunikace na jiných portech, např při TLS port 587 nebo SSL port 465 bývá průchozí, protože při takových přenosech se počítá s nutností autorizace na cílovém SMTP serveru a nejedná se tedy o Open Relay.

Další možností je částečné omezení SMTP spojení v závislosti na počtu odeslaných zpráv za daný čas (hodinu, den, jednotlivou zprávu) nebo velikostí zprávy (od 2MB do 10MB).

2.2.2 Obrana založená na SMTP spojení

Když je obrana na straně odesílatelů neúspěšná a rozesílatelům spamu se podaří odeslat zprávy, je nevyhnutelné rozpoznávat správnou poštu (někdy uváděnou jako „ham“) od spamu na straně cílových domén, serverech. K tomu slouží následující techniky [11].

Black/White/Greylisty

Mezi nejstarší systém pro zjišťování spamu lze považovat tzv. černé seznamy nebo listiny, často označované jako DNSBL (Domain Name System Black List). Jedná se o seznam známých rozesílatelů spamu založený na IP adrese nebo doménovém jménu odesílajícího e-mailového serveru. Postup vyhledávání v databázi je takový:

1. Na cílový poštovní server dorazí zpráva. Zjistí se IP adresa odesílatele.
2. Následně se obrátí pořadí jednotlivých bajtů adresy a připojí se k doménovému (FQDN) jménu odesílatele.
3. Vygenerovaný řetězec se odešle do DNSBL serveru, který vrátí odpověď.
4. Jestliže se vrátí odpověď, která je adresa typu loopback, odesílatel není na seznamu. V opačném případě se pravděpodobně jedná o systém rozesílající spam.

V případě vyhledávání podle domény, je žádost složená s domény druhé úrovně místo IP adresy a FQDN serveru.

Podskupinou blacklistů jsou seznamy obsahující adresy open relay serverů a otevřených proxy serverů, které mají za úkol zamaskovat skutečnou IP adresu odesílatele.

Bílé seznamy mají obdobnou funkci jako černé s tím rozdílem, že obsahují IP adresy nebo doménová jména serverů, kterým se odesílání e-mailů povolí.

Speciálním druhem seznamu je tzv. šedý neboli Greylist. Využívá se tehdy, když současná protispamová ochrana nedokáže jednoznačně identifikovat příchozí zprávu. Jedná se o několik technik, z nich si popíšeme jednu navrženou Evanem Harrisem.

Je využito IP adresy serveru od kterého přišla zpráva, e-mailové adresy odesílatele a e-mailové adresy příjemce. Těmto informacím se říká tzv. triplet a slouží jako identifikátor poštovní relace. Většina spamu je tvořena zprávami od původců, kteří ještě s cílovým adresátorem nikdy nekomunikovali. Rozhodování je provedeno na základě tripletu. Pokud již byla někdy tato komunikace navázaná, považuje se zpráva jako „ham“, tedy v pořádku. Pro neznámé relace je využita technika pozdrženého přijetí e-mailové zprávy popsané dále.

Pozdržení příjetí zprávy - Tarpitting

Při vytváření poštovních protokolů se počítalo s případem, kdy cílový adresát nemusí být okamžitě dostupný. V takovém případě se přepravní uzel opakovaně snaží dopravit poštu. Naproti tomu, je situace odesílání nevyžádané pošty. Zde jde o nejrychlejší odeslání co nejvíce pošty v nejkratším možném čase. Případná opakovaná odeslání nepřipadají v úvahu. Tohoto principu se snaží využívat techniky pozdržení přijetí zprávy.

Zahajovací navázání komunikace mezi odesílatelem a příjemcem je záměrně prodlouženo. Odesílatel musí počkat až příjemce spojení potvrdí a následně může zprávu odeslat. U běžné pošty s tímto nebývá problém, narozdíl od odesílání spamu. Zde je důležité v co nejkratším čase odeslat co nejvíce pošty.

Ověřování odesílatelů

Mezi nejrozšířenější techniky pro ověřování odesílatelů patří SPF (Sender Policy Framework. Do DNS TXT (Domain Name System Text Record) záznamu se uvedou ip adresy serverů, které mohou odesílat poštu pro danou doménu. Při příjmu pošty stačí pomocí DNS dotazu zjistit pověřené odesílatele a pokud je v obálce uveden jiný než na seznamu tak je možné zprávu zahodit[21].

Nevýhodou je zvětšení zátěže na DNS servery a mobilní uživatelé, kteří se často přpojují z různých sítí (lze řešit připojením do firemní sítě).

Podobnou technikou založenou na asynchronní kryptografii je DKIM (DomainKeys Identified Mail). Při odeslání pošty server zprávu digitálně podepíše, přijímající server naopak podpis ověří veřejným klíčem umístěným v DNS TXT záznamu odesílatele. Nevýhoda tohoto způsobu je vysoká náročnost na zpracování a nutnost ověřování celé zprávy.

2.2.3 Obrana založená na obsahu zpráv

Bývá výpočetně náročnější než při kontrole SMTP spojení. Z toho důvodu je také zařazovaná až jako jedna z posledních kontrol.

Bayesova analýza

Je založena na aplikování statistického modelování na jakékoliv textové řetězce ohraničené mezerou. Tyto řetězce je možné shromažďovat i pomocí různých OCR řešení, kdy se obrázek převádí na textovou informaci. Rozhodovací proces musí být založen na určitém základu správných a nevyžádáných zpráv. Těmto zprávám se říká trénovací a musí být u nich rozhodnuto ručně. Z takto získaného základu se již Bayesovy filtry učí předpovídat typ zprávy.

Distribuované sítě

Pokud uživateli e-mailové schránky dorazí mail a označí jej jako nevyžádaný, vytvoří se kontrolní součet zprávy. Ten je odeslán do distribuované antispamové sítě. Kontrolní součty jsou generovány pomocí tzv. fuzzy kontrolních součtů, které umožňují akceptovat drobné rozdíly ve zprávách.

Jakmile dojde nová zpráva, server vytvoří kontrolní součet, který se odešle do databáze sítě. Odpověď obsahuje vyjádření, zda-li zpráva je ham nebo spam.

2.3 Nebezpečí v poštovních zprávách

Jak již bylo psáno na začátku kapitoly, často se v nevyžádané poště objevují nakažlivé infekce. Ty jsou nebezpečné pro koncové uživatele a můžou způsobovat různé druhy zákeřností v závislosti na typu nákazy. V našem případě se jedná o vytváření sítí ovládaných útočníkem za účelem rozesílání spamu, tzv. botnety. Sílu těchto ovládaných sítí dokazuje ukončení činnosti botnetu Rustock 15. března 2011. Týden po odstavení bylo celkové množství spamu o 40,4% menší [26].

2.3.1 Viry

Obdobně jako v biologii je počítačovým virem napadnut spustitelný soubor (exe, vbs, atd.), který slouží jako hostitel. Aby vir mohl fungovat a dál se šířit, musí být daný soubor spuštěný. Někdy se pojem vir zaměňuje za jakoukoliv škodlivou aplikaci na počítači. Jediná možnost šíření virů je předáváním spustitelných souborů přes síť nebo vyměnitelným médiem (USB disk, atd.).

2.3.2 Červy

Jedná se o nákazu využívající zranitelnosti cílových stanic, kdy se zcela automaticky šíří pomocí sítí na další stroje. Existují dva druhy červů, první typ má za úkol vyřadit síť z provozu pouze svým šířením (SQL Slammer využívá chyby přetečení zásobníku při dotazování na Microsoft SQL Server). Opakem je druhý typ, ten má za úkol provádět nechtěné operace na poškozené stanici (např. vytváření botnetů, mazání souborů atd.). Z poslední doby se jedná o červ Conficker.

2.3.3 Trojské koně

Většinou se jedná pouze o spustitelný soubor, který se nedokáže sám rozšiřovat. Maskuje se za užitečné aplikace, ale neparazituje na nich jako u virů. Šíření trojských koní je nejčastěji přes e-maily nebo stahováním software z neověřených stránek. Jakmile se troský kůň dostane na počítač, tak se samovolně dále nešíří. Některé trojské koně se vydávají za antivirové řešení, ale na rozdíl od nich do počítače nahrávají škodlivý software. Účelem trojských koní je získat kontrolu nad zasaženou stanicí a umožnit její vzdálené ovládání (stanice se stane součástí tzv. botnetu).

3 TVORBA E-MAILOVÉHO SERVERU

Cílem práce je vytvořit poštovní serveru běžící na operačním systému GNU/Linux, dále v textu bude používáno zkráceně pouze Linux. Jedná se o operační systém vytvořený Linusem Torvaldsem a komunitou kolem projektu GNU. Není zpoplatněn a společnosti, které vyvíjí distribuce pro komerční využití mají založený finanční model na poskytování podpory pro zákazníky. Existuje mnoho druhů distribucí specializujících se na serverové nebo uživatelské využití. Linux společně s celým projektem GNU je založen na softwaru typu open source. Jeho výhodou je možnost nahlédnutí do zdrojových souborů a možnost úpravy podle vlastní potřeb.

3.1 Debian

Debian je distribuce vytvářená společností Debian Project. Jejím primárním cílem je využití na serverech. Umožňuje běh na mnoha architekturách od nejběžnějších i386 nebo x86-64 až po ARM či SPARC. Charakteristickou vlastností Debianu je jeho softwarová konzvervativnost, kdy se používají ověřené verze softwaru v testovacích větvích než se dostanou do nasazení pro ostrý provoz. Jednotlivé větve se jmenují unstable, testing a stable, která je použita v této práci.

3.1.1 Použitý hardware

Použitý hardware je počítač obsahující 32 bitový procesor Pentium 4 s 512 MB paměti RAM. Z tohoto důvodu byla verze Debianu vybraná i386, která se hodí pro danou architekturu PC. Instalace byla provedena ze spouštěcího CD typu Netinst.

To obsahuju samotný operační systém s minimem programů, vše co budeme potřebovat se doinstaluje ze sítě. Takový přístup k instalaci je vhodný z důvodu, že neinstalujeme software, který nebudeme potřebovat. To nám umožňuje vytvořit systém méně náchylný k chybám v programech, jež nepotřebujeme. Bylo použito aktuální sestavení, Debian Lenny r6, dostupné z adresy http://www.debian.org/distrib/netinst.

3.1.2 Instalace operačního systému

Instalace je velice jednoduchá a skládá se z několika kroků. Po spuštění bootovacího CD zvolíme instalaci. Na začátku je potřeba vyplnit obecné údaje jako jsou jazyková lokalizace, rozvržení klávesnice a fyzické umístění serveru. Následuje ověření instalačního CD a prověření komponent počítače. Jakmile je vše v pořádku je požadováno nastavit doménové jméno (titan) a doménu (jirifrantisek.cz), plně kvalifikované doménové jméno (FQDN) je tedy titan.jirifrantisek.cz.

Dalším krokem je rozdělení disku. Toto nastavení nemá žádnou funkci na provoz poštovního serveru, zvolíme tedy použití celého disku s jediným oddílem, který bude obsahovat všechny systémové složky. Po provedení diskových operací je nainstalován základní systém. Následně je vhodné vytvořit heslo pro uživatele root a vytvoření běžného uživatele.

Po výzvě vybereme softwarové zrcadlo ze kterého se budou stahovat aktualizace a programy. Je vhodné vybrat z některých fyzicky blízkých zrcadel, aby byla rychlost přenosu co nejrychlejší. Několik je umístěno i v české republice. Aby bylo možné instalovat software ze sítě, je nutné nastavit připojení k serveru proxy. Pokud žádný nepoužíváme, můžeme nechat prázdné.

V předposledním kroku vybereme pouze základní systém a počkáme na doinstalování zbylých balíků. Úplně na závěr potvrdíme zapsání zavaděče GRUB do záznamu MBR disku. Tím je umožněno automatické spouštění operačního systému po zapnutí počítače. Jakmile se objevý výzva, vytáhneme instalační CD a restartujeme systém.

3.1.3 Nastavení sítě

Po startu systému dostal počítač IP adresu z lokálního DHCP serveru. Tento způsob je vhodný pro běžné počítače, ale méně pro servery. U nich si musíme být jistí na jaké adrese komunikují s okolím. Krátký výpadek serveru DHCP může způsobit veliké problémy, proto raději nastavíme IP adresu pevnou. To se provede editací souboru /etc/network/interfaces pomocí jakéhokoliv textového editoru (obvykle můžeme použít vi nebo nano). Výsledná konfigurace může vypadat takto:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 10.0.0.1
netmask 255.255.255.0
network 10.0.0.0
broadcast 10.0.0.255
gateway 10.0.0.138
```

První část udává lokální rozhraní, tzv. lokální smyčku. Druhá část již specifikuje síťové rozhraní eth0 a nastavuje parametry vhodné pro naši síť (hodnotu IP adresy, masky, výchozí brány, volitelně může obsahovat IP adresu sítě a adresu broad-

castu). Po vložení příkazu `/etc/init.d/networking restart` dojde k restartování síťového subsystému a provede se načtení upravených nastavení.

Debian při běžném nastavení vrací příkazem hostname pouze doménové jméno. Editací souboru /etc/hostname můžeme nastavit plné FQDN jméno.

```
echo titan.jirifrantisek.com > /etc/hostname
```

Při překladu jména na IP adresu je nejdříve prohlédáván soubor /etc/hosts, následně systémová cache s DNS záznamy a jako poslední se provádí dotaz na DNS server. Upravíme soubor hosts, aby překládal doménové jméno na správné adresy:

```
127.0.0.1 localhost.localdomain localhost
10.0.0.1 titan.jirifrantisek.cz titan
```

Nyní zbývá restartovat příkazem `/etc/init.d/hostname.sh start` službu hostname a nastavení překladu na lokální systém je dokončeno.

3.2 Postfix

Postfix je přenosový agent MTA, který protokolem SMTP přenásí zprávy od uživatelů (MUA) nebo vzdálených serverů na další servery. Je ho možné provozovat na nejrůznějších UNIX-ových systémech od Linuxu přes MacOS X až po BSD nebo Solaris. Sendmail je dalším MTA používaných pod Linuxem, ze kterého Postfix vychází a je s ním zpětně kompatabilní. Vývoj Postfixu, oproti vývoji Sendmailu, byl zaměřen více na bezpečnost, modularizaci procesů a možnost snadnější konfigurace.

3.2.1 Instalace MTA

Instalace většiny software v Debianu je velice snadná. Využívá se balíčkového systému dpkg, který se stará o závislosti na jiných balíčcí apod. Potřebné balíčky pro běh systému nainstalujeme:

```
apt-get install postfix postfix-doc
```

V interaktivním okně instalátoru vybereme: `No configuration`. Postfix se nastavuje pomocí dvou hlavních souborů umístěných ve složce /etc/postfix. První soubor se jmenuje master.cf a slouží k nastavení démona master, který spouští služby podle potřeby. K celkové konfiguraci Postfixu slouží druhý konfigurační soubor main.cf. Ten můžeme upravovat v textovém editoru nebo příkazem `postconf`

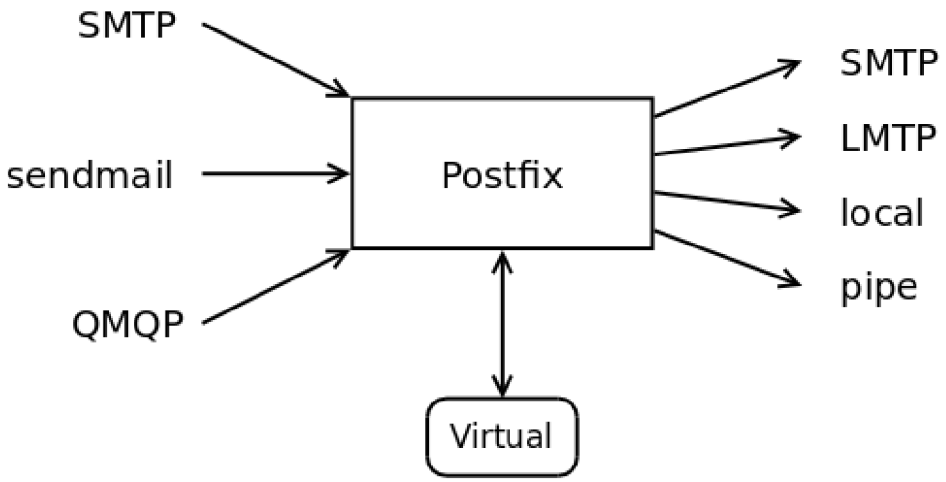
`-e „příkaz"`. Základní konfigurace by mohla vypadat:

```
# FQDN jméno
myhostname = titan.jirifrantisek.cz
# příjem zpráv pro domény
mydestination = titan.jirifrantisek.cz, localhost
# nastavení místní sítě
mynetworks = 127.0.0.0/8
```

Jedná se o základní nastavení, které pouze ověří správnost instalace Postfixu. Příkaz myhostname upravuje doménové jméno na plně kvalifikované. Další řádek specifikuje cílové domény pošty zpracovávané Postfixem. Nastavení umožňuje přijímat poštu pouze pro lokální uživatele systému (viz. následující kapitola). Parametr mynetworks slouží pro nastavení omezení možnosti odesílat poštu na adresu lokální smyčky, tedy odesílat pouze ze serveru. Po každé editaci konfiguračních souborů je důležité provést příkaz „postfix reload" pro načtení nových nastavení.

3.2.2 Vyhledávací mapy

Jakmile přijde zpráva do MTA, je nutné se rozhodnout co se s ní provede. Obdobně jako při směrování paketů. Na rozdíl od routerů a jejich směrovacích tabulek slouží u Postfixu k tomuto účelu tzv. mapy. Ty nám říkají co se provede se zprávou přijaté na jednom rozhraní a na jeké rozhraní se bude směrovat. Více popisuje obr.3.1.



Obr. 3.1: Vyhledávací mapa virtual rozhoduje o přesměrování ze vstupu na výstup

Vyhledávací mapy můžou mít různou podobu. Může se jednat o indexovaný binární soubor vytvořený z běžného textového souboru nebo se Postfix může dotazovat

na LDAP nebo do databázového stroje. Vstupem do systému je nejčastěji relace s jiným serverem komunikujících na různých protokolech. Postfix při svém rozhodování může doručovat zprávy do těchto výstupů:

- local - Slouží k ukládání pošty pro systémové uživatele na lokálním serveru (root, atd.).
- smtp - Přeposílá poštu do vzdálených systémů.
- virtual - Jedná se o omezenou verzi démona local s možností ukládání pošty do lokálního uložiště uživatelům bez nutnosti přihlašování k operačnímu systému serveru.
- lmtp - Předává poštu lokálním schránkovým serverům.
- pipe - Rozhraní pro odesílání pošty do jíných transportních mechanismů (fax, pomocné programy, atd.).

3.3 MySQL

MySQL je databáze založená na SQL jazyku a je šířena pod licencí GPL. Je možné ji provozovat na nejrůznějších systémech. Hlavní doménou stále zůstává nasazení v konfiguraci LAMP poháněné linuxem. Existuje celá řada databází ať už komerčních nebo ne, např. Microsoft SQL, IBM DB2, Oracle Database[1]. Mezi ne SQL databáze patří např. CouchDB vyvíjená společností Apache Software Foundation.

MySQL byla považována vždy za jednodušší databázi, ale v posledních verzích přibyla řada důležitých vlastností, které ji pomohli srovnat krok z konkurencí (např. funkce trigger, transakce, atd.).

Databáze slouží pro ukládání dat a jejich vyhledávání. Na rozdíl od běžného ukládání dat do souboru databáze přináší řadu výhod:

- Sdílení - lepší řešení přístupu více uživatelů k datům.
- Bezpečnost - pokročilé nastavování práv uživatelům.
- Integrita dat - je možné nastavit vztahy mezi daty, které není možné zničit.

MySQL poskytuje několik možných uložišť dat, označované jako engine. Základní dva typy jsou InnoDB a MyISAM. InnoDB se více hodí pro tabulky, do kterých se zapisují důležité údaje. Má velkou režii a je i pomalejší než MyISAM, ale její hlavní výhodou je možnost transakcí nad operacemi. Každá operace se ukládá do

[1]Oracle je po akvizici se společností Sun Microsystems vlastníkem databáze MySQL

logu a pokud vznikne neočekávaná akce (např. násilné vypnutí systému, nedostatek volného místa) tak je možné vrátit původní stav dat.

MyISAM vyniká výkonem zejména při čtení z databáze[2]. Indexy je možné ukládat do fyzický jiných uložišť než je samotná tabulku a navíc má pokročilejší možnosti indexace sloupců. Indexace, je možnost rychlejšího vyhledávání hledané hodnoty, než lineární procházením tabulky [8].

3.3.1 SQL jazyk

Nejčastějším druhem databází jsou databáze založené na SQL jazyku. SQL je dorozumivací jazyk mezi člověkem a databázovým strojem, který je tvořen dotazem a následnou odpovědí. Jedná se o neprocedurální jazyk, uživatel tedy nemusí definovat jak data získat. Dotaz je možné tvořit čtyřmi základními příkazy[5]:

- SELECT - čtení dat
- INSERT - vložení dat
- UPDATE - změna dat
- DELETE - smazání dat.

3.3.2 Instalace databáze

Z důvodu jednodušší správy cílových domén a jejich poštovních uživatelů byla zvolena možnost ukládat informace o schránkách do databáze. K doinstalování databáze MySQL do Debianu byl opět využit balíčkovací systém dpkg. Po zadání příkazu:

```
apt-get install postfix-mysql mysql-client mysql-server libpam-mysql
```

Se nás systém zeptá na heslo pro uživatele root do databáze. Tento uživatele je podobně jako v operačním systému správcem celé databáze.

3.3.3 Konfigurace databáze

Nejprve je důležité vytvořit samotnou databázi, do které se budou ukládat data o doménách a uživatelých. To se provede příkazem: `mysqladmin -u root -p create mail` Vytvoření uživatele s opravnéním pouze na naši databázi je vhodné z bezpečnostního hlediska. Pokud by se útočník dostal do databáze, nemohl by provádět manipulace s všemi databázemi. Přihlášení do konzole MySQL je možné provést

[2]zápis je také rychlejší

příkazem (`mysql -u root -p`). Vytvoření nového uživatele mail_admin s heslem „tajne_heslo“ se provede [24]:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON mail.* TO
'mail_admin'@'localhost' IDENTIFIED BY 'tajne_heslo';
FLUSH PRIVILEGES;
```

Databáze obsahuje tabulky, které slouží pro ukládání parametrů o uživatelích, doménách, přesměrováních:

- users - Zde jsou uloženy e-mailové adresy uživatelů a jejich hesla pro přístup ke schánce v šifrované podobě. Hesla jsou šifrována pomocí funkce `ENCRYPT`.
- domains - Obsahuje doménové adresy, pro které přijímá systém poštovní zprávy.
- forwardings - Přesměrování z e-mailových adres na další e-mailové adresy je uloženo v tabulce forwardings. Zde se může nastavit i tzv. doménový koš.
- transports - Podobně jako u tabulky forwardings tak i zde se nastavuje přesměrování. S tím rozdílem, že se jedná o celé domény.

3.3.4 Komunikace Postfixu s MySQL

Pro komunikaci Postfixu s MySQL jsou použity soubory, které specifikují dotazy na databázi. Jednotlivé dotazy se budou měnit v závislosti na požadované funkci a Postfix si je bude automaticky volat. Každý soubor bude obsahovat přihlašovací údaje, název a adresu do databáze s příslušným SQL dotazem. Příkladem může být soubor zjišťující, zda-li existuje adresát pošty [31]:

```
user = mail_admin
password = tajne_heslo
dbname = mail
hosts = 127.0.0.1
query = SELECT CONCAT(SUBSTRING_INDEX(email,'@',-1),'/',
SUBSTRING_INDEX(email,'@',1),'/') FROM users WHERE email='%s'
```

Většina dotazů získává jednoduché údaje (např. doména, cílová adreasa přesměrování, atd.). Pouze mysql-virtual_mailboxes.cf, uvedený v příkladu, provádí rozdělení adresy na uživatelskou část a doménu oddělenou znakem /. Nastavení Postfixu pro komunikaci s MySQL se provádí v /etc/postfix/main.cf např:

```
virtual_mailbox_maps = proxy:mysql:/etc/postfix/mysql-
virtual_mailboxes.cf
```

Proměnné začínající znaky „proxy:“ slouží pro zlepšení přístupu do vyhledávací mapy. Každé vyhledání je zpřístupněno přes démona proxymap, který zmenšuje počet spojení, v konkrétním případě do databáze. Navíc obchází omezení při použití změny kořenového adresáře (chroot). Všechny vyhledávací tabulky, které jsou otevírány pomocí démona proxymap je potřeba definovat v příkazu proxy_read_maps nebo proxy_write_maps.

Jelikož je použit režim virtuálních schránkových domén, který používá účty bez ohledu na lokální uživatele uvedených v /etc/passwd, je nutné definovat přesměřování (tzv. alias) jinak než v souboru /etc/aliases. Můžeme ho provést příkazem virtual_alias_maps, který nastavuje cílové domény. Vyhledávací mapa s e-mailovými adresami je nastavena pomocí virtual_mailbox_maps.

3.3.5 Uložiště zpráv

Ukládání poštovních zpráv je možné provést mnoha způsoby. Byl zvolen systém ukládání do struktury adresářů maildir. Datová struktura uložiště bude umístěna v domovském adresáři uživatele vmail a bude jejich jediným vlastníkem. Vytvoření skupiny a přidání nového uživatele je možné provést pomocí příkazů:

```
groupadd -g 5000 vmail
useradd -g vmail -u 5000 vmail -d /home/vmail -m
```

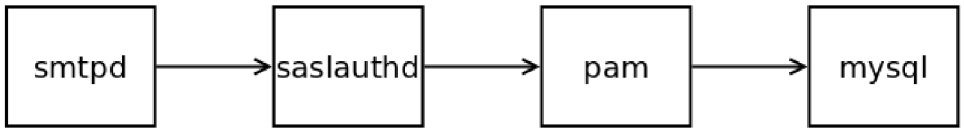
Číslo 5000 je identifikátor uživatele a skupiny (tzv. uid a gid). Domovský adresář je uložen v /home/vmail. Nastavení v Postfixu provedeme přidáním řádků do main.cf:

```
virtual_mailbox_base = /home/vmail
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000
```

Nastavení nám říká, kam se bude pošta ukládat a s jakými právy (uživatele a skupiny).

3.4 Nastavení SASL

Jelikož nechceme mít ze serveru Open Relay je nutné omezit možnost odesílání pošty. Dále budeme předpokládat, že je potřeba odesílat poštu i z neznámých sítí. Z tohoto důvodu se používá autentifikace uživatelů, kdy si server při odesílání pošty ověřuje pravost odesílatele. Ověření uživatele bude probíhat pomocí protokolu SASL. Přihlašovací informace budou posílány v běžném textu. Z důvodu zvětšení bezpečnosti bude použito TLS pro zašifrování přenosového kanálu, tím i hesla při přenosu. Více v kapitole o TLS na straně 17. Ověřování bude probíhat systémem zobrazeným na obr.3.2. Klient bude chtít odeslat e-mail pomocí smtpd, ten mu odpoví, že je nutné se autorizovat. Klient pošle přihlašovací údaje na saslauthd, ten je předá pro lokální ověření v pam a podle dat v databázi se rozhodne o přidělení oprávnění.



Obr. 3.2: Princip autentifikace uživatelů

Abychom mohli šifrovat, musíme vygenerovat věřejný a soukromý certifikát. Můžeme k tomu využít programu OpenSSL:

```
openssl req -new -outform PEM -out smtpd.cert -newkey rsa:2048
-nodes -keyout smtpd.key -keyform PEM -days 365 -x509
```

Po zodpovězení několika osobních informací (jméno, firma, adresa atd.), jsou vygenerovány certifikáty v aktuálním adresáři (uložíme je do /etc/postfix. Opět upravíme konfiguraci Postfixu v souboru main.cf a přidáme tyto řádky:

```
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_mynetworks,
permit_sasl_authenticated, reject_unauth_destination
smtpd_use_tls = yes
smtpd_tls_cert_file = /etc/postfix/smtpd.cert
smtpd_tls_key_file = /etc/postfix/smtpd.key
```

První příkaz spouští ověřování uživatelů pomocí SASL. Další umožňuje ověřování

klientů, kteří nedodržují striktně standarty [3]. Příkaz smtpd_recipient_restrictions nastavuje omezení pro odesílatele. V tomto případě se jedná o možnost odeslat poštu pouze ze sítě nastavené v mynetwork a uživatelům, kteří se ověří pomocí SASL. Navíc odmítá požadavky pokud Postfix není předávající nebo cílové místo. Poslední nastavení povolují podporu TLS a udávají cesty k veřejným a privátním certifikátům.

Nyní musíme nainstalovat balíky pro běh SASL:

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

SASL reprezentuje démon saslauthd, jeho hlavní konfigurační soubor je uložen /etc/default/saslauthd. V tomto souboru nastavíme spouštení po startu změnou příznaku „START“ na hodnotu „yes“ a změnu parametru „OPTIONS“ na hodnotu OPTIONS=”-c -m /var/spool/postfix/var/run/saslauthd -r”. Je to z důvodu změny cesty k pojmenovanému soketu [4] na kterém poslouchá ověřovací mechanismus. Parametr „-r“ udává spojování uživatelské části e-mailové adresy s doménovou částí adresy. Nový adresář vytvoříme příkazem:

```
mkdir -p /var/spool/postfix/var/run/saslauthd
```

Saslauthd provádí komunikaci s e-mailovým klientem. Jelikož saslauthd neumí ověřovat uživatelské údaje uložené v databázi MySQL, tak musí být použit PAM (Pluggable Authentication Modules). Jeho nastavení je uloženo v souboru /etc/pam.-d/smtp [30], ten by měl obsahovat údaje pro nastavení komunikace s databází [30]:

```
auth required pam_mysql.so user=mail_admin passwd=tajne_heslo
host=127.0.0.1 db=mail table=users usercolumn=email
passwdcolumn=password crypt=1
account sufficient pam_mysql.so user=mail_admin passwd=tajne_heslo
host=127.0.0.1 db=mail table=users usercolumn=email
passwdcolumn=password crypt=1
```

Na závěr je potřeba nakonfigurovat démona smtpd, aby využíval SASL. To je uloženo v souboru /etc/postfix/sasl/smtpd.conf:

```
pwcheck_method: saslauthd
```

[3] Některé starší verze Microsoft Outlooku a Outlook Expressu [3]

[4] Soket označuje spojení IP adresy a portu

```
mech_list: plain login
allow_plaintext: true
auxprop_plugin: mysql
sql_hostnames: 127.0.0.1
sql_user: mail_admin
sql_passwd: tajne_heslo
sql_database: mail
sql_select: select password from users where email = '%u'
```

Pro správný běh SASL přidáme uživatele postfix do skupiny sasl a restartujeme obě služby:

```
adduser postfix sasl
/etc/init.d/postfix restart
/etc/init.d/saslauthd restart
```

3.5 POP/IMAP server

Z důvodu snadné konfigurace a dobré spolupráce s Postfixem byl pro stahování pošty klientů použit Courier Mail Server [13].

3.5.1 Instalace Courier

Po nainstalování potřebných balíků příkazem:

```
apt-get install courier-authdaemon courier-authlib-mysql courier-pop
courier-pop-ssl courier-imap courier-imap-ssl
```

Provedeme konfiguraci Courieru. Jeho konfigurační soubor pro ověřování uživatelů z databáze je umístěn v /etc/courier/authmysqlrc. Měl by vypadat takto [13]:

```
MYSQL_SERVER localhost
MYSQL_USERNAME mail_admin
MYSQL_PASSWORD tajne_heslo
MYSQL_PORT 0
MYSQL_DATABASE mail
MYSQL_USER_TABLE users
MYSQL_CRYPT_PWFIELD password
```

```
MYSQL_UID_FIELD 5000
MYSQL_GID_FIELD 5000
MYSQL_LOGIN_FIELD email
MYSQL_HOME_FIELD "/home/vmail"
MYSQL_MAILDIR_FIELD CONCAT(SUBSTRING_INDEX(email,'@',-1),'/',
SUBSTRING_INDEX(email,'@',1),'/')
```

Konfigurace nastavuje připojení k databázi a parametry databáze, které slouží pro ověření uživatelů. Zbytek souboru určuje údaje potřebné pro ukládání pošty (práva pro ukládání, cestu kam se budou zprávy ukládat, atd.). Dále je nutné ověřit správné nastavení autentizačního démona. Je uložen v /etc/courier/authdaemonrc a pro ověřování v MySQL by mělo být nastaveno „authmodulelist="authmysql"“. Pokud budeme chtít využívat zabezpečené komunikace pomocí SSL, je nutné vygenerovat certifikáty.

Nastavení generování certifikátů je pro jednotlivé protokoly uloženo v souborech /etc/courier/imapd.cnf a /etc/courier/pop3d.cnf, které upravíme podle potřeby. Můžeme měnit typ certifikátu a identifikační údaje. Samotné generování certifikátů se provede pomocí příkazů:

```
mkimapdcert
mkpop3dcert
```

Nyní už zbývá pouze restartovat služby, které jsme upravovali:

```
/etc/init.d/courier-authdaemon restart
/etc/init.d/courier-imap restart
/etc/init.d/courier-imap-ssl restart
/etc/init.d/courier-pop restart
/etc/init.d/courier-pop-ssl restart
```

3.5.2 Vytvoření uživatelů

Vytvoření uživatelů probíhá naplněním tabulek v databázi. Nejprve je potřeba specifikovat doménu, pro kterou bude Postfix přijímat poštu [5]:

```
INSERT INTO ‘domains‘ (‘domain‘) VALUES ('jirifrantisek.cz');
```

[5] Struktura databáze je v příloze.

Dále již můžeme přidávat jednotlivé uživatele. Stačí uvést e-mailovou adresu a heslo pro přístup k účtu:

```
INSERT INTO ‘users‘ (‘email‘, ‘password‘) VALUES
(’jirka@jirifrantisek.cz’, ENCRYPT(’heslo’));
```

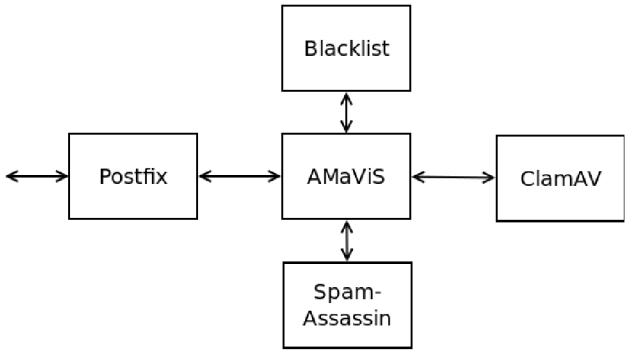
Pro potřeby přesměrování pošty slouží tabulka forwardings. Nejjednodušším způsobem přesměrování je z jedné adresy na druhou:

```
INSERT INTO ‘forwardings‘ (‘source‘, ‘destination‘)
VALUES (’jiri@jirifrantisek.cz’, ’jirka@jirifrantisek.cz’);
```

Poštu z jednoho účtu můžeme přesměrovat na více účtů pokud je oddělíme čárkou. Další možností přesměrování je vytvoření tzv. „catch-all“ adresy, která bude přijímat poštu pro všechny neexistující uživatele. Můžeme ji vytvořit nastavení proměnné source na hodnotu např. @jirifrantisek.cz a destination na jiri@jirifrantisek.cz.

4 VYTVOŘENÍ ANTISPAMOVÉHO FILTRU

Antispamový filtr bude tvořený třemi prvky. Prvním prvkem bude program Amavis, ten bude mít za úkol filtrovat příchozí poštu z Postfixu, připravovat ji pro další analýzy a předávat pro antivirovou, antispamovou kontrolu. Bude se tedy jednat o externí filtr pošty pro Postfix. Dalšími prvky budou programy pro kontrolu došlých zpráv, antivirová kontrola bude prováděna programem ClamAV a antispamová kontrola pomocí SpamAssassin. Vztahy mezi jednotlivými prvky jsou uvedeny na obr.4.1.



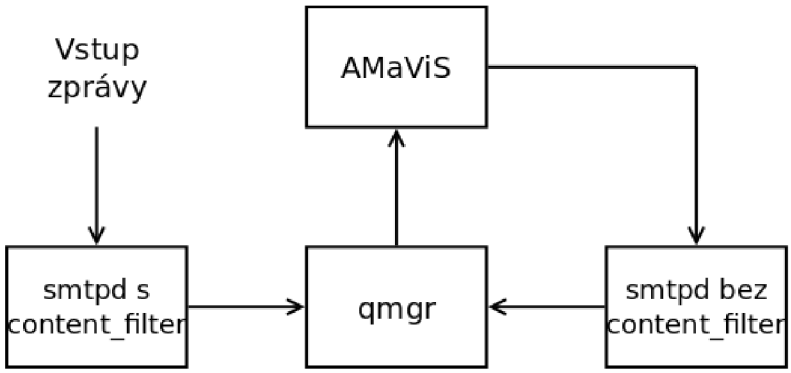
Obr. 4.1: Vazby Amavisu s ostatními částmi systému

4.1 Amavisd-new

Pro filtrování pošty bude použit program Amavisd-new, který vychází z projektu Amavis (A Mail Virus Scanner), v následujícím textu bude uváděno zkráceně Amavis. Jedná se o rozhranní mezi MTA a antispamovou či antivirovou kontrolou. Je napsán v jazyku Perl, což zaručuje přenositelnost mezi operačními systémy. S MTA komunikuje pomocí SMTP, LMTP (Local Mail Transfer Protocol) nebo jinými pomocnými programi. Pokud je provozován s programem SpamAssassin a přijatá zpráva obsahuje více příjemců bude provedeno testování zprávy pouze jednou. Při instalaci poštovního systému na více serverů je vhodné umístit Amavis na stranu tzv. hraničního serveru (z anglického edge server), který přijímá poštu jako první, bez ohledu na umístění uživatelských schránek. V systému bude plnit funkci přebírání zpráv z Postfixu, následné předávání prvkům antispamové ochrany a přípravu příloh pro antivirovou kontrolu [18].

Postfix umožňuje definovat externí filtry, které po průchodu vstupní frontou předají zprávy jiným aplikacím. Přesměrování je realizováno SMTP démonem s nastaveným content_filtrem (externím filtrem) a může být směrováno např. do Amavisu.

Využití s dvěma smtp démony je znázorněné na obr.4.2. Jeden slouží pro vyjmutí zprávy z Postfixu a druhý ji vrací zpět. Filtr zpráv může komunikovat pomocí smtp, lmtp, pipe a důležité je, aby jeden z smtpd neobsahoval content_filter, protože by došlo k zacyklení zpráv.



Obr. 4.2: Integrace Amavisu do Postfixu

Amavis a další podpůrné balíčky nainstalujeme příkazem:

```
apt-get install amavisd-new zoo unzip bzip2 nomarch lzop pax
libnet-ph-perl libnet-snpp-perl libnet-telnet-perl
```

Spolu s hlavními částmi programu jsou instalovány podpůrné perl-ovské knihovny a archivační nástroje pro rozbalování zkomprimovaných e-mailových příloh. Všechny konfigurační soubory jsou uloženy v adresáři /etc/amavis/conf.d/. Odkomentování řádků začínajících na @bypass_virus_checks_maps a @bypass_spam_checks_maps v souboru 15-content_filter_mode zapneme přesměrování zpráv z Amavisu do antivirové a antispamové kontroly. Do souboru 50-user přidáme $pax='pax'; (nesmí být na posledním řádku [12]). Jedná se o specifikaci uživatelského rozhranní aplikace paxutils.

Nastavení filtru content_filter provedeme v souboru main.cf přidáním řádků:

```
content_filter = amavis:[127.0.0.1]:10024
receive_override_options = no_address_mappings
```

Nejdříve specifikujeme filtr, jeho jméno, umístění na lokálním počítači a daném portu [1]. Dalším krokem je vytvoření a úprava smtpd v souboru master.cf. Nakonec souboru přidáme:

```
amavis unix - - - - 2 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
```

[1] nastavení portu je možné změnit v souboru `/etc/amavis/conf.d/20-debian_defaults`

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

První část nastavuje přenos zpráv do Amavisu. Jedná se o definici smtp klienta se jménem amavis (musí být stejné jméno jako je uvedeno v content_filter v souboru main.cf). Bylo nastaveno omezení na maximálně dva SMTP klienty a prodloužen časovač pro potvrzení přijetí zprávy.

Druhá část vytváří smtp démona, který bude přijímat zprávy z Amavisu do Postfixu. Nejdůležitější z parametrů je prázdná hodnota content_filter, která vypíná filtr a zprávy tedy nebudou končit v nekonečné smyčce, ale budou procházet do démona qmgr.

Je důležité upravit soubor 05-domain-id, aby obsahoval správnou definici proměnné @local_domains_acl. Ve výchozí konfiguraci se přebírá název domény ze souboru /etc/mailname, kde se uvádí celé doménové jméno serveru. Pro správnou funkci při hostování více domén je nutné tyto domény uvést do závorky, oddělené čárkou a domény musí být v uvozovkách. Špatná konfigurace může vést k problémům v komunikaci mezi Amavisem a SpamAssassinem.

4.2 ClamAV

ClamAV je open source antivirus vytvořený za účelem prověřování e-mailových zpráv. Primárně je pro systémy na bázi UNIXu [2]. Může paralelně zpracovávat několika démony velké množství zpráv. Poskytuje rozhranní pro příkazovou řádku a obsahuje pokročilý systém aktualizece virových definic. Má zabudovanou podporu pro soubory typu MS Office, HTML, RTF a PDF. Instalaci provedeme příkazem:

```
apt-get install clamav clamav-daemon
```

[2] V současné době je dostupný i jako běžný antivirový systém pro operační systém Microsoft Windows

Konfigurace spočívá pouze v přidání uživatele clamav do skupiny amavis a restartování služeb.

```
adduser clamav amavis
/etc/init.d/amavis restart
/etc/init.d/clamav-daemon restart
/etc/init.d/clamav-freshclam restart
postfix reload
```

V souboru /etc/amavis/conf.d/15-av_scanners se konfigurují antivirové programy pro kontrolu zpráv. Pro správnou funkci ClamAV je nutné mít nastaveno proměnnou @av_scanners, @av_scanners_backup a @bypass_virus_checks_maps.

ClamAV se skládá ze dvou služeb, jedna je samotný program běžící jako démon a druhá je clamav-freshclam, která slouží pro aktualizace virové databáze [14].

Amavis si při startu detekuje nainstalované externí kompresní programy, antiviry a antispamová řešení. To můžeme ověřit prohlédnutím systémového logu [3]:

```
amavis[3312]: Found decoder for .tar at /usr/bin/pax
amavis[3312]: Using primary internal av scanner code for ClamAV-clamd
```

4.3 SpamAssassin

Jedná se o antispamové řešení, které podrobuje příchozí zprávu testům s velkým množstvím aktualizovaných pravidel. Každé pravidlo má svoji hodnotu, která je přičtena při splnění pravidla ke zprávě. Při průchodu testy jsou jednotlivé výsledky sečteny. Pokud zpráva přesáhne stanovenou mez, může být označena za spam nebo dokonce úplně smazána.

Nastavení limitů při rozhodování o druhu zprávy je nutné nastavit v konfiguračním souboru Amavisu 20-debian-defaults, jelikož SpamAssassin pouze kontroluje zprávy, ale vyhodnocení je otázkou Amavisu. Existují tři úrovně spam tag level, spam tag2 level a spam kill level. Při překročení první se přidávají pouze hlavičky X-Spam, druhá úroveň mění předmět zpráv a přidává na jeho začátek řetězec `***SPAM***`. Pokud má zpráva hodnocení větší než třetí úroveň tak dojde k jejímu smazání nebo uložení do karantény. Pravidla ve filtrech se zaměřují na:

- Analýzu hlavičky zprávy
- Analýzu těla zprávy
- Bayesovu analýzu
- Černé seznamy DNS

[3] Celkový výpis logu byl zkrácen

- Databáze kontrolních součtů
- Další specifické pravidla

V dalším textu je rozebráno téma databází kontrolních součtů a jako specifické pravidlo je uvedeno Fuzzy OCR. Instalace spočívá v nainstalování balíčku spamassassin z repozitářů. Konfigurace je uložena v několika souborech. Systémová nastavení specifická pro danou verzi jsou uložena v souborech např. v310.pre, uživatelská v souboru local.cf. Toto rozdělení je z důvodu aktualizací SpamAssassinu, aby nedocházelo k přepisování uživatelských pravidel při aktualizaci. Spuštění Bayesova filtrování provedeme editací souboru `local.cf`.

```
use_bayes 1
use_bayes_rules 1
bayes_auto_learn 1
```

Po každé úpravě SpamAssasssinu je potřeba ověřit konfiguraci pomocí `spamassassin --lint`, nemělo by vypsat žádnou chybu. Aktualizace pravidel se provádí příkazem `sa-update --no-gpg`, aby se nemusela provádět ručně, nastavíme ji jako naplánovanou úlohu do crontabu. Příkazem `crontab -e` můžeme přidávat naplánované úlohy. Vložením řádku `30 2 */2 * * /usr/bin/sa-update --no-gpg &> /dev/null` se budou pravidla aktualizovat každý druhý den v 2:30 ráno.

4.3.1 Databáze kontrolních součtů

Razor

Celým jménem Vipuls Razor, jedná se o program a distribuovanou síť, která slouží jako databáze spamových zpráv. Pokud označíme e-mail jako spam, je z něj vytvořena značka a odeslána do sítě. Při příchodu zprávy je porovnávána značka s databází v síti. Hash je tvořen pomocí fuzzy logiky, kdy jsou povoleny drobné rozdíly mezi zprávami. Pokud je zpráva označena jako spam je možné s ní provádět mnoho věcí, od přidání bodového ohodnocení pro SpamAssassin po změnu předmětu, přidání hlavičky atd [11].

Nahlašování spamu do sítě Razor je omezeno reputací přispívajícího uživatele. Pokud někdy nahlásí falešný mail může mu být omezena možnost nahlašování. Zakladatelé sítě také varují před automatickým nahlašováním spamu pomocí automatizovaných systémů (SpamAssasssin, Bayesovy filtry). Nejvhodnější je hlásit spam, který byl roztřízen lidmi. Instalace se provede nainstalováním balíčku:

```
apt-get install razor
```

Nyní se nastaví SpamAssassin, aby používal Razor. Následující řádky se přidají do souboru `/etc/spamassassin/local.cf`:

```
use_razor2 1
razor_config /etc/razor/razor-agent.conf
```

Pyzor

Stejně jako Razor tak i Pyzor je distribuovaná síť se svojí klientskou a serverovou částí. Ze zpráv se vytvářejí tzv. otisky a ty se porovnávají s databází na serveru podobně jako u Razoru. Pyzor původně vznikal jako implementace Razoru v programovacím jazyku Python. Při vývoji se autorům nelíbila implementace komunikačního protokolu a softwarová uzavřenost Razor serverů.

Jelikož je vytvoření otisku z celé zprávy náročná operace na výpočetní výkon mailových serverů, je generování hashe pouze z části zprávy [28]:

- Hlavičky zpráv se zahazují.
- Odstraňují se textové řetězce delší než deset znaků.
- Odstraňují se z textu e-mailové adresy, url adresy, html značky a prázdné znaky.
- Pokud je zpráva delší než čtyři řádky, vybírají se z ní pouze některé části.

Nainstalování je obdobné jako u Razoru:

```
apt-get install pyzor
```

Editací `/etc/spamassassin/local.cf` povolíme Razor:

```
use_pyzor 1 pyzor_path /usr/bin/pyzor
```

DCC

DCC je zkrtka pro Distributed Checksum Clearinghouse. Stejně jako Pyzor je DCC také antispamová síť, která je open source projekt a poskytována zdarma. Narozdíl od Razoru je způsob ověřování spamu postaven na četnosti výskytu nahlášení dané zprávy. Čím více uživatelů nahlásilo zprávu jako spam, tím je větší pravděpodobnost, že se jedná o nevyžádanou zprávu. Každý den je v síti DCC ověřováno přes 300 milionů zpráv [29]. Instalace se musí provést kompilování zdrojových kódů, protože Debianí repozitáře neobsahují instalační balíček.

```
cd /tmp
wget http://www.dcc-servers.net/dcc/source/dcc-dccproc.tar.Z
tar xzvf dcc-dccproc.tar.Z
cd dcc-dccproc-1.3.102
./configure --with-uid=amavis
make
```

```
make install
chown -R amavis:amavis /var/dcc
ln -s /var/dcc/libexec/dccifd /usr/local/bin/dccifd
```

4.3.2 Fuzzy OCR

Jedná se o přídavný modul do SpamAssassinu, který dokáže analyzovat obrázkové e-maily [16]. K rozpoznávání správné pošty od nevyžádané používá následující techniky:

- OCR (Optical Character Recognition) neboli rozpoznávání znaků v obrázku pomocí externích algoritmů (gocr, ocrad).
- Fuzzy logiku na výsledek OCR analýzy.
- Vytváření hashů z obrázků a ukládání do databáze (MySQL, MLDBM).

Byla provedena instalace poslední verze 3.6.0, která je vedená jako testing, nicméně je plně funkční. Stažení provedeme příkazem:
`wget http://users.own-hero.net/ẽcoder/fuzzyocr/fuzzyocr-3.6.0.tar.gz`
Rozbalíme stažený archiv a všechny soubory začínající na FuzzyOcr.* a adresář FuzzyOcr je nutno překopírovat do adresáře spamassassina. Ten si příslušné konfigurační soubory (*.cf) sám načtě při vyvolání amavisem. Pro správný chod je nutné ze sítě CPAN doinstalovat moduly String::Approx, Time:Hires, MLDBM::Sync, Log::Agent, dále balíčky Ocrad, Netpbm, Gifsicle, Libungif-bin.

4.4 Černé seznamy DNS

Funkce Blacklistů byla již popsána v kapitole 2.2.2, zde uvedu pouze některé seznamy. Všechny jsou používány SpamAssassinem pro označování zpráv. Více jich je uvedeno na *http://www.declude.com/Articles.asp?ID=97*.

4.4.1 SpamCop

Do seznamu SpamCop se rozesílatel spamu dostane pouze nahlášením některého z uživatelů. Záznam je uložen na určitou dobu v závislosti na poměrovém bodovacím systému. Množství bodů se odvíjí od doby jak dlouho je již adresa na seznamu, jestli je spam nahlášen jako první nebo již byla přijata stejná zpráva. Čím dříve je spam nahlášen po přijetí, tím větší bodové ohodnocení dostane zdroj. Dalším možným nahlášením je zachytávání zpráv do tzv. spamových pastí. Jedná se o e-mailové adresy, které jsou vytvořeny pouze za účelem přijímání spamu. Nejsou nikde

zveřejňovány, ani jakýmkoliv způsobem využívány. Pošta, která do nich dorazí se považuje za spam. Po 48 hodinách od posledního nahlášení zprávy se záznam smaže ze seznamu.

4.4.2 DSBL

Jedná se o zkratku ze slovního spojení Distributed Server Boycot List, neboli distribuovaný seznam bojkotovaných serverů. Tvoří ho skupina administrátorů a uživatelů, kteří se spojili v boji proti spamu. Obsahuje několik seznamů v závislosti podle uživatelů, kteří do seznamu přispívali. Rozlišuje se na tzv. důvěryhodné a nepotvrzené uživatele. Do důvěryhodných seznamů mohou přispívat pouze zaregistrovaní uživatelé. Naproti tomu do nepotvrzených se může přispívat anonymně. Pokud bude důvěryhodný uživatel nahlašovat legitimní zprávy, tak mu může být členství zrušeno.

4.4.3 Spamhaus

Je založen na myšlence, že 90% nevyžádané pošty v Evropě a Severní Americe je rozesíláno z méně než 200 známých tvůrců spamu. Úkolem administrátorů Spamhausu je sledovat tyto tvůrce a zjišťovat jejich stěhování od jednoho poskytovatele síťových služeb k druhému. Aktualizace seznamu se provádí přibližně jednou za hodinu.

4.4.4 NJABL

Neboli Not Just Another Bogus List (žádný další falešný seznam). Je provozován poštovními administrátory, kterým nevyhovují ostatní černé listiny. Skládá se pouze z jednoho seznamu, který obsahuje jakoukoliv IP adresu pokud splňuje tyto podmínky:

- Jedná se o otevřený poštovní server nebo otevřený proxy server.
- IP adresa leží v rozsahu dynamicky přidělovaných adres.
- Bylo zjištěno rozesílání nevyžádané pošty z této IP adresy.

Služba je poskytovaná zdarma a jen je nutné se přihlásit k odběru elektronického zpravodaje. Přispět pro rozvoj seznamu může každý administrátor, který využívá jeho služeb. Stačí si změnit připojovací metodu, kdy se bude odesílat každá IP adresa, která se chce připojit k serveru, na otestování jestli se nejedná o otevřený poštovní server.

5 WEBOVÁ APLIKACE

Jako v každém odvětví tak i v informačních technologií jsou trendy, které udávají směr vývoje. Do 80-tých let byla infrastruktura tvořena sálovými (mainframeovými) počítači. Jednalo se o hlavní výpočetní kapacitou pro tzv. tenké klienty, jejichž úkolem bylo pouze zobrazovat výstup a umožnit vstup k chodu programu běžícím na hlavním stroji.

Následoval nástup osobních počítačů, kdy hlavním cílem zvyšování výpočetního výkonu byl založen na frekvenci procesoru. Trvalo dlouhou dobu než se trend začal opět obracet. Procesorová architektura NetBurst používaná firmou Intel začala s nástupem 21. století dosahovat svých fyzikálních limitů, kdy bylo velmi problematické překročit pracovní frekvence procesorů přes 4GHz se zachováním rozumného ztrátového tepla.

Vývoj se přestal zabývat co nejrychlejším zpracováním sériových dat, ale přešel k paralelismu informací. To vede k přesouvání dat a výpočetního výkonu do novodobých sálových počítačů pod pojmem tzv. cloudových služeb. Uživatel vůbec netuší, kde jsou data fyzicky zpracovávána. Na Internetu se začínají objevovat aplikace v právém slova smyslu. Může se jednat o různé komunikační služby, sociální sítě nebo až po dostupnost celých kancelářských balíků a her provozovaných v internetových prohlížečích. Webové aplikace se skládají z určitých prvků a protokolů, které jsou rozebrány v následujících kapitolách.

5.1 Apache

Webový server je prvek sloužící pro komunikaci s klienty prostřednictvím protokolu HTTP nebo zabezpečenou verzí HTTPS. Nejrozšířenějším webovým servrem je Apache, obsluhuje požadavky přibližně na 60% doménách z celého světě ([25]). Nejčastěji se používá v tzv. LAMP kombinaci stejně jako v této práci. Zkratka je odvozena od prvních písmem slov Linux, Apache, MySQL, PHP, více o PHP v kap.5.4.1. Jedná se o open-source tvořený skupinou dobrovolníků. Existuje velké množství dalších webových serverů, např. Light HTTP Server nebo IIS pro operační systém Windows [2].

Mimo provozování HTTP a HTTPS serveru umožňuje Apache vytvářet virtuální domény, reverzní proxy sloužící pro rozdělení zátěže, atd. Samozřejmostí je podpora skriptovacích jazyků.

5.1.1 Instalace a konfigurace Apache

Apache je v repozitářích uložen v balíčku pod názvem apache2. Po nainstalování jsou konfigurační soubory uloženy v adresáři /etc/apache2, konkrétně se jedná o apache2.conf a httpd.conf. Ve výchozím stavu jsou všechny dokumenty ve složce /var/www dostupné z Internetu a budou Apachem předávány pomocí HTTP protokolu.

Pro správné fungování PHP je potřeba doinstalovat balíčky libapache2-mod-php5, php5, php5-mysql.

5.2 HTML

HTML je značkovací jazyk sloužící k tvorbě webových dokumentů. Jeho úkolem je říci klientskému prohlížeči jak má přijatý dokument vypadat. K tomu slouží značky, které říkají co má jak vypadat. Doslova revoluční myšlenkou bylo použití odkazů do různých částí dokumentu. Procházení textu je tedy nelineární např. od čtení knih a vede k dřívějšímu zjištění informací.

V současné době je nejrozšířenější verzí 4.01, ale již se začíná nasazovat verze 5, která je stále zatím vedena jako nedokončený standard (Working Draft). Nová verze je zaměřena mimo jiné na multimediální obsah a její implementace by měla nahradit proprietární software jako je Microsoft Silverlight, Adobe Flash Player, atd.

5.3 CSS

Úkolem kaskádových stylů je oddělit obsah od formy. Vzhled dokumentu je definován odděleně a pomocí značek (tagů) je nastavováno jak daný text uvnitř značky bude vypadat. Výhodou je možnost rozdělení programového kódu na tzv. front-end a back-end. Back-end připravuje hodnoty pro front-end, který je prezentuje uživateli.

Každý prvek dokumentu má značku a ta má definovaný vzhled. Tímto způsobem je možné lehce změnit v celém dokumentu např. vzhled nadpisů první úrovně. Další možností je přizpůsobování celých stránek požadavkům uživatelů a vytváření tzv. skinů. Záleží pak na každém, jaký si zvolí.

5.4 Skriptovací jazyky

Slouží pro vytváření skriptů, určitých dávkových souborů, které není nutno překládat do strojového kódu. Jedná se tedy o podskupinu interpretovaných programovacích jazyků. Jejich výhodou je tedy snadná změna zdrojového kódu bez nutnosti

kompilace. Nevýhodou bývá nižší rychlost vykonávání programů (skriptů) oproti kompilovaným programovacím jazykům.

5.4.1 PHP

PHP je rekurzivní zkratka pro PHP: Hypertext Preprocessor, který je vyvíjen jako open source. Nejčastěji se používá pro tvorbu dynamických internetových stránek stejně jako v této práci. Jeho syntaxe je podobná jazyku C a proto je blízká velkému množství programátorů. Velice dobře spolupracuje s webovým serverem Apache a obsahuje velmi dobrou komunikaci s databázemi, hlavně s MySQL.

Zdrojový kód je prováděn na straně serveru a klientovi jsou poslány pouze výsledky ve formě HTML. Zdrojové soubory se ukládají s příponou *.php, aby server věděl, že má požadovaný dokument zpracovat jako skript php.

5.4.2 Perl

Jedná se o velice kontroverzní jazyk, který není vhodný pro začínající programátory. Na druhou stranu je vytvořen pro psaní efektivního kódu s konkrétním účelem bez zbytečností. Nejčastěji je mu vytýkána absence deklarování proměnných, nelze deklarovat vlastní datové typy a syntaxe není zcela striktní. Hlavní oporou jsou asociativní pole a regulární výrazy [6].

V perlu jsou napsány programy Amavis, Spamassassin a jejich konfigurační soubory jsou psány v perlovské syntaxi. Mnoho modulů je uloženo v síti CPAN (Comprehensive Perl Archive Network) a je možné je doinstalovávat podobně jako instalační balíčky do Debianu.

5.5 Bezpečnost přístupu k souborům a programům

Běžná konfigurace poštovního serveru spolu s antispamovým řešením je prováděna editací konfiguračních souborů. Je tedy nutné zajistit oprávnění pro přístup k těmto souborům a některým příkazům (např. pro restartování běžících služeb). Webový server Apache je provozován pod uživatelem www-data, jak můžeme vyčíst z výstupu běžících služeb:

```
www-data 30863 24520 3896 S 06:25 0:00 /usr/sbin/apache2 -k start
```

Jeho oprávnění jsou pro chod celého serveru velice omezené. Musíme tedy zajistit, aby byly skripty prováděny s dostatečným oprváněním. Existuje mnoho řešení jak takového cíle dosáhnout, v této práci budou použito programu Sudo.

5.5.1 Sudo

Jeho úkolem je umožnit spouštění vybraných příkazů uživatelům nebo skupinám s vyšším oprávněním než, které mají přiřazené. Příkaz sudo využívá bezpečnostní politiku uloženou v souboru /etc/sudoers nebo v LDAP. Každé použití přikazu sudo je zaznamenáváno do logů a je požadováno ověřením heslem uživatele, který sudo chce spustit (na rozdíl od přikazu su, kdy se musí vložit přihlašovací údaje uživatele cílového). Pokud uživatel spouští příkaz s cílovým oprávněním stejným jako on sám, není potřeba zadávat heslo. Z toho plyne, že výzva heslem není vyžadována ani u uživatele root. Ověřování heslem je možné i vypnout.

Sudoers

Soubor sudoers je používán jako nejčastější místo, kde se ukládají pravidla pro uživatele. Obsahuje tři druhy údajů tzv. aliasy, pravidla a režijní příkazy. Řežijní příkazy mění chovádní příkazu sudo (nastavení logování, doba uložení hesla, atd.). Aliasy se nemusí používat, ale značně zpřehledňují výslednou konfiguraci. Existují čtyři druhy aliasů:

- User_Alias - pro vytvoření uživatelských skupin bez ohledu na systémové skupiny
- Runas_Alias - umožňuje přebírat oprávnění jiných skupin
- Host_Alias - definuje stanici podle doménového jména nebo ip adresy
- Cmnd_Alias - vytváří skupiny příkazů

Syntaxe psaní v souboru sudoers vychází z rozšířené Backhus-Naurovy formy (EBNF). Vypadá následovně:

```
Cmnd_Alias PAGERS = /bin/cat /etc/postfix/mysql-virtual*.cf
Cmnd_Alias SERVICES = /etc/init.d/amavis restart
www-data ALL = NOPASSWD: SERVICES, PAGERS
```

Nejdříve jsou nadefinovány dva aliasy, které se použijí v posledním řádku. Ten říká: Uživatele www-data na všech stanicích (v tomto případě pouze jeden server) může spouštět příkazy bez hesla uvedené v aliasech SERVICES a PAGERS (pozn. názvy aliasů musí být velkými písmeny).

5.6 Úvodní stránka

Slouží pro přihlášení uživatelů do systému správy antispamového řešení. Na pravé straně je přihlašovací formulář pro vstup do systému. Pro správné přihlášení je potřeba použít přihlašovací údaje k e-mailovému účtu. Existují tři role uživatelů:

- Lokální administrátor - může nastavovat pouze údaje týkající se svého účtu.
- Doménový administrátor - může nastavovat pouze údaje vybrané domény.
- Globální administrátor - správa všech účtů.

Pro uložení rolí uživatelů je využita tabulka users. Ta se využívá hlavně pro vytváření e-mailových adres a nyní je rozšířena i pro udržování rolí. Jednotlivé role jsou po načtení z databáze uchovávány v proměnné $_SESSION. Ošetření vstupních polí je zabezpečeno funkcí mysql_real_escape_string() a navíc je kontrolován formát adresy.

Obr. 5.1: Úvodní stránka aplikace

Celá aplikace se skládá z mnoho skriptů, dají se rozdělit na několik částí. První slouží k tvorbě stránky jako takové. Zde se generuje navigační lišta, postraní panel, atd., uložení těchto skriptů je v adresáři ./pag_div. Druhou skupinou jsou pomocné skripty, ty obsahují funkce, které se vyskytují napříč aplikací. Např. se jedná o naplnění roletek z databáze, ověřování hodnot ze vstupů. Poslední částí jsou skripty uložené v adresáři scripts. Zde jsou nastaveny přihlašovací údaje k databázi a funkce zajišťující zablokování přístupu nepřihlášených uživatelů ke stránkám, které vyžadují přihlášení. Pohyb mezi stránkami je řešen pomocí formulářů a metody POST sloužící pro uchovávání předaných hodnot.

5.7 Příprava Amavisd-new

5.7.1 Ukládání proměných

Konfigurace Amavisu je rozdělena do dvou základních adresářů. Vyhodnocování začíná nejdříve v adresáři /usr/share/amavis/conf.d, zde jsou uloženy základní nastavení. Pro uživatelská nastavení slouží /etc/amavis/conf.d. Oba adresáře obsahují soubory začínající číslem, to určuje pořadí vyhodnocování v daném uložišti. Hodnota uložená v posledním souboru při procházení konfigurací má přednost před svými výskyty v předchozích souborech, tzn. soubory s vyššími čísly mají větší prioritu. Konfigurační soubory jsou zdrojové soubory programovacího jazyku Perl, ve kterém je Amavis naprogramován. Drží si tedy perlovskou syntaxi přistupování, definování proměnných a volání funkcí.

Konfigurační nastavení je možné ukládat do proměnných typu:

- konstanta
- hash tabulka (asociativní pole)
- ACL (běžné pole)

Rozdíl mezi hash tabulkou a ACL je ve způsobu ukládání a vyhledávání dat. Hash je efektivnější pro větší množství dat a může vracet různé hodnoty mimo pravda, nepravda. ACL je jednodušší a lineárně procházené.

5.7.2 Vyhledávání v proměnných

Vyhledávání je realizováno pomocí tzv. map. Všechny proměnné končící *_maps jsou vyhledávací mapy, definice takové mapy vypadá [19]:

`@virus_admin_maps = (\%virus_admin, \$virus_admin)`. Vyhledávání může probíhat v běžných proměných (konstanta, pole, hash) a navíc v databázích SQL nebo LDAP. Speciálním případem je vyhledáváním v regulárních výrazech. Ty podrobují hledanou hodnotu zadaným testům uvnitř regulárního výrazu. Používá se například při blokování zakázaných příloh, kde se testuje koncovka přílohy.

5.7.3 Vyhledávání nad SQL tabulkou

Vyhledávání v tabulkách nejčastěji používá jako vstupní proměnou e-mailovou adresu. Ta je určitým způsobem měněna a vyhledávání je prováděno v několika krocích.

Zástupný znak + je nastaven proměnnou $recipient_delimiter a je možné ji specifikovat. Důležité je umístění znaku zavináče při vyhledávání nad doménami. Na rozdíl od vyhledávání v asociativních polích je zavináč připojován k doménové části

adresy. Vyhledávání se ukončí při první kompletní shodě některého z vyhledáváných výrazů. Postup prohledávání je následující:

1. jmeno+prijmeni@jirifrantisek.cz
2. prijmeni@jirifrantisek.cz
3. jmeno+prijmeni
4. jmeno
5. @jirifrantisek.cz
6. @.jirifrantisek.cz
7. @.cz
8. @.

5.8 Banky politik

Jakmile přijde zpráva z Postfixu na vstupní rozhranní démona Amavis tak je podrobena kontrole. Rozsah této kontroly je definován v konfiguračních souborech Amavisu. Antispamová kontrola je rozdělena do několika kroků:

- Kontrola SMTP hlaviček.
- Kontrola na zakázané přílohy.
- Antivirová kotrola.
- Antispamová kontrola.
- Kontrola černých/bílých seznamů.

Politiky jdou měnit globálně pro všechny přijaté zprávy nebo je možné využít banky politik (policy bank). Banky se využívají při nutnosti nastavení rozdílných testů pro uživatele, domény, apod. K tomuto problému je možné přistupovat dvěma způsoby.

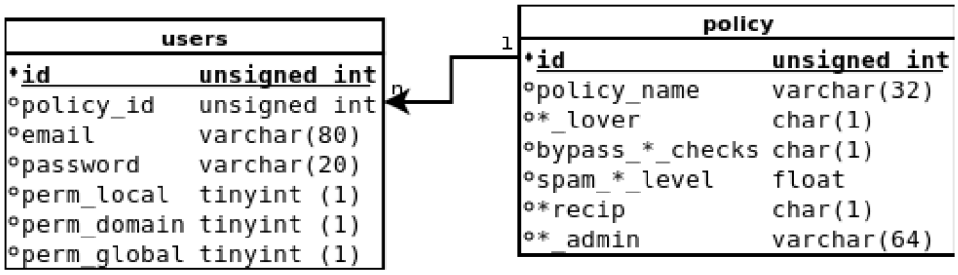
První způsob spočívá v nastavení konkrétní banky na démona Amavisu. Těchto démonů může být aktivních více a záleží na filtrování postfixu podle (smtpd_sender/recipient_restrictions), který přesměrovává poštu na konkrétního démona.

Způsob využitý v této diplomové práci je založený na uložení politik v databázi. Každý uživatel má nastavenou politiku, která se aplikuje při kontrole zprávy. Tento způsob umožňuje jemněji nastavit politiky a navíc je celá situace přehlednější, protože se vše nastavuje na jednom místě.

Komunikace Amavisu s databází je řešena pomocí databázového rozhraní DBI a perlovského modulu DBD::mysql. Připojení k databázi je udržováno v proměnné @lookup_sql_dsn, do které se ukládá spojení s databází vytvořenou příkazem DBI:mysql. Připojení je uložené v souboru 50-user. Prohledávání v politikách je podle e-mailové adresy příjemce, konkrétně se jedná o SQL příkaz SELECT z tabulky politik.

5.8.1 Databázový model politik

Pro aplikování politik budou stačit pouze dvě tabulky. Tabulka, která je již využívána systémem Postfix pro uložení uživatelů, je rozšířena o atribut policy_id. Ten slouží pro nastavení politiky danému uživateli.



Obr. 5.2: ER diagram politik.

Tabulka policy obsahuje kromě názvu a id politiky položky, které ovlivňují chování systému. Názvy dalších položek nejsou zvoleny náhodně, ale odpovídají konfiguračním hodnotám Amavisu. Některé jsou tvořeny textem nebo čísly. Položky obsahující jeden znak (char(1)) mají troj-stavovou logiku. Můžou obsahovat hodnoty Y, N nebo null. Pokud je nastaveno na null, tak je hodnota nenastavená a použije se výchozí nastavení systému. Celá struktura databáze je uvedena v příloze a vychází z dokumentace [20]. Zprávy mohou být:

- zavirované
- spam
- se zakázanou přílohou
- obsahující špatné hlavičky

Hodně nastavení je společných pro vyjmenované čtyři typy zpráv. V obr.5.2 byly záznamy zkráceny a nahrazeny hvězdičkou. Každý tedy existuje pro čtyři druhy zpráv. Konkrétní nastavení ovlivňují:

- *_lovers - Projde k adresátovi i když je označena za vadnou.

- bypass_*_checks - Kontrola se nebude vůbec provádět.
- *recip - Varovat odesílatele o nedoručení zprávy.
- *_admin - Varování admina o přijetí závadné zprávy.

Zprávy označené jako spam mají větší množství nastavení. Definují se úrovně se zvláštním významem. Je možné nastavit hladiny těchto úrovní:

- spam_tag_level - přidávání hlaviček kontrol
- spam_tag2_level - možný spam, do předmětu se vkládá text z hodnoty spam_subject_tag.
- spam_kill_level - odeslání do karantény
- spam_dsn_cutoff_level - odesílání zprávy o doručení (DSN - Delivery status notification)
- spam_quarantine_cutoff_level - přímé smazání zprávy

5.8.2 Aplikování politiky

Při příchodu zprávy do Amavisu se začnou aplikovat nastavení postupně podle místa uložení a názvu souborů. Nejdřív se začíná v umístění /usr/share/amavis/, následuje /etc/amavis/conf.d/. V Debianu je konfigurace rozdělována do několika souborů. Každý soubor začíná číslem a procházení je vždy od nejnižších hodnot po nejvyšší. Z toho plyne, že soubory s nejvyšším číslem mají nejvyšší prioritu a při shodě nastavení se aplikuje výskyt v posledním souboru.

Proměnná @lookup_sql_dsn nastavuje připojení k databázi za účelem vyhledávání (v politikách,černých seznamech, atd.). Dotaz na databázi pro výběr politiky je uložen v proměnné $sql_select_policy:

```
SELECT *,users.id FROM users,policy WHERE (users.policy_id=policy.id)
AND (users.email IN (%k))
```

Úkolem tohoto příkazu je vybrat všechny hodnoty proměnných z tabulky policy pro uživatele uloženého parametru „%k“, kde reference policy_id na politiku je cizím klíčem v tabulce users. Způsob zpracování parametru „%k“ je uveden v kapitole 5.7.3.

5.8.3 Přiřazení politiky uživatelům

Jednotlivé politiky jsou spravovány na kartě „Politiky“. Zde se vypisují uživatelé v závislosti na přiděleném oprávnění a je možnost jim přiřazovat vytvořené politiky. Je možné politiky také vytvářet nebo měnit. Oprávnění mají za úkol pohlídat vztah

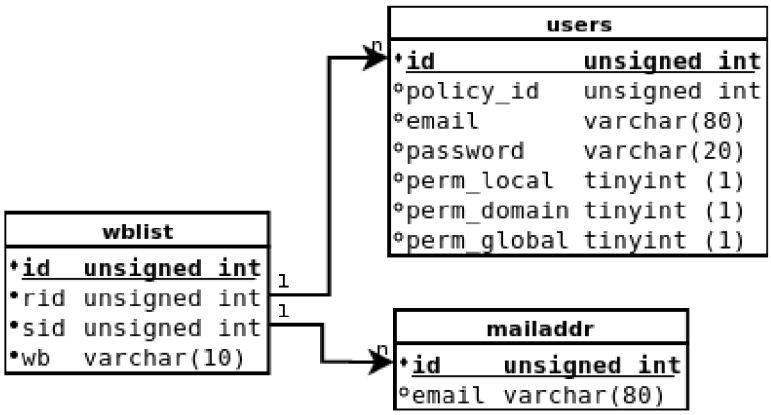
Obr. 5.3: Přehled uživatelů a přiřazených politik

mezi rolí uživatele a možností měnění politiky. Nesmí docházet k situacím, kdy si uživatel s právem lokálního administrátora změní politiku, která se aplikuje i na jiné osoby.

Při zmáčknutí tlačítka „Vytvořit“ nebo „Změnit“ je uživatel přesměrován na stránku edit_pol.php. Zde je možné upravit vybranou politiku nebo ji smazat. Formulářová data jsou ošetřována proti sql injektion funkcí mysql_real_escape_strings().

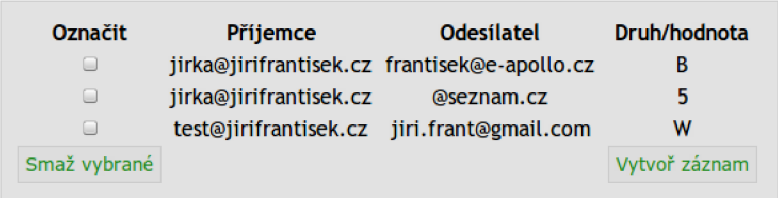
5.9 Uživatelské černé a bílé seznamy

Uživatelské černé a bílé seznamy jsou ukládány do tabulky wblist 5.4. Její obsah tvoří cílový adresát (majitel seznamu - rid), odesílatel (blokovaná nebo upřednostňovaná položka - sid) a druh seznamu (wb). Poslední uvedený atribut je možné nastavit pomocí znaků W a B nebo číselnou hodnotou, která se přičítá k výslednému hodnocení SpamAssassinu. V případě použití znaku W, B nedochází k testování zprávy SpamAssassinem, ale zpráva je okamžitě propuštěna nebo odeslána do karantény.



Obr. 5.4: ER diagram černých a bílých seznamů.

Hodnota sid je cizím klíče do tabulky mailaddr, která je tvořena položkou email. Mailaddr je možné plnit jak celou e-mailovou adresou nebo také jen její částí. Výsledný vzhled seznamů je tvořen příkazem SELECT z databáze a uložen do proměnné $sql_select_white_black_list definované v souboru 50-user. Vyhledávání nad sql tabulkou je podrobněji popsáno v kapitole 5.7.3. Celá struktura databáze je uvedena v příloze a vychází z dokumentace [20].

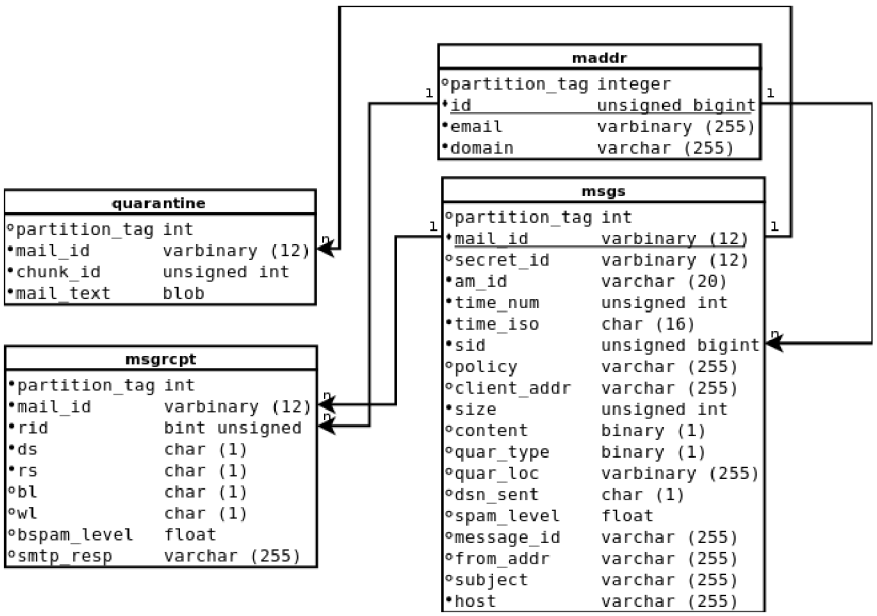


Obr. 5.5: Zobrazení černých a bílých seznamů.

5.10 Karanténa

Pokud je poštovní zpráva spam nebo je jakýmkoliv způsobem Amavisem označena za závadnou (odesílatel je na černé listině, nevhodná příloha, atd.) je uložena do karantény. Ve výchozím nastavení jsou zprávy komprimovány a ukládány do složky /var/lib/amavis/virusmails. Z důvodu lepší spolupráce s webovou částí antispamové administrace, bylo zvoleno uložiště pro karanténu v databázi.

Podobně jako při vyhledávání v sql tabulkách, tak i pro zápis je nutné definovat rozhranní do proměnné @storage_sql_dsn. Zápis má stejnou syntaxi jako pro @lookup_sql_dsn. Může se využít stejných přihlašovacích údajů, ale je možné využít úplně



Obr. 5.6: ER diagram tabulek karantény.

odlišných, včetně jiného druhu databáze.

Na obr.5.6 je zobrazen ER diagram, sloužící pro ukládání zpráv do karantény. Návrh databáze je použit z oficiální dokumentace Amavisu [20]. Všechny tabulky obsahují proměnnou partition_tag. Jejím hlavním cílem je zpřehlednit data uložená v databázi a usnadnit její čištění. Ovlivnit její hodnotu můžeme proměnnou $sql_partition_tag.

Zprávy jsou ukládány do sloupce blob v tabulce quarantine. Tento typ uložiště slouží pro ukládání převážně binárních dat, zde byl použit pro zachování formátu zprávy. Amavis přiřadí každé zprávě dva indentifikátory mail_id a secret_id. První je veřejný identifikátor dané zprávy a druhý se používá pro vyzvednutí zprávy z karantény. Secret_id není uváděn do žádných logů a je ukládán pouze do databáze.

V tabulce msgs jsou uloženy informace o odesílateli (e-mailová a ip adreasa), vlastnosti zprávy (velikost, předmět, hlášení o doručení). Tabulka msrcpt obsahuje data o příjemci zprávy (status doručení, jestli byl odesílatel na černém nebo bílém seznamu příjemce, atd.).

5.10.1 Vyzvednutí z karantény

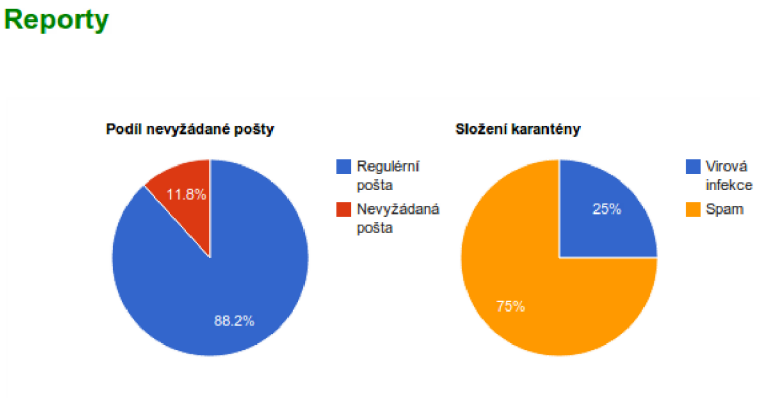
Zprávu z karantény je možné ručně doručit příjemci pomocí programu `amavisd-release`. Pomocí identifikátorů mail_id a secret_id je vybrána zpráva a odeslaná do postfixu bez dalších kontrol. V grafickém rozhranní byl přístup k programu řešen pomoci příkazu sudo v kap.5.5. Všechny zprávy v karanténě jsou zobrazeny na stránce

quarantine.php. Dohledávání zpráv je ulehčeno pomocí řazení nad vybranými.

5.11 Reporty

Výsledky kontrol nad zprávami jsou ukládány do tabulky msgs. Zde se uvádí, zda-li je zpráva odeslána do karantény, případně z jakého důvodu. Tyto údaje jsou využívány pro přehledné zobrazení uživtelům. Pro tvorbu grafů bylo použito rozhranní Google Chart Tools dostupné z webové stránky http://code.google.com.

Rozhranní je naprogramováno v JavaScriptu. Jedná se o interpretovaný programovací jazyk podobně jako PHP s tím rozdílem, že vykonávání kódu je prováděno na straně klienta [10]. Použití Google Chart Tools je velice jednoduché. Načtení rozhranní je provedeno funkcí `google.load`, následně se ve funkci `drawChart` nastaví proměnné grafů, které jsou vyčteny z databáze pomocí PHP. Samotný objekt koláčového grafu je vytvořen konstruktorem `google.visualization.PieChart`. Vykreslení proběhne pomocí metody `draw`, které se předají doplňující parametry (data, rozměry a název grafu).



Obr. 5.7: Vzhled reportů.

6 ZÁVĚR

V první fázi práce byla rozebrána problematika přenosu zpráv elektronické pošty s používanými protokoly. Na to navázala praktická realizace poštovního serveru Postfix, běžící na operačním systému GNU/Linux. Klienti mohou stahovat poštu prostřednictvím protokolů POP3, IMAP a jejich zabezpečených verzí pomocí TLS. Při odesílání pošty přes SMTP je ověření uživatele prováděno pomocí SASL a stejně jako u příjmu pošty je komunikace šifrovaná protokolem TLS. Pošta je ukládána do souborů typu Maildir. Uživatelské účty s doménami, které přijímá Postfix jsou uloženy do MySQL databáze se zašifrovanými hesly.

Cílem diplomové práce bylo popsta a navrhnout antispamového řešení. K tomuto účelu bylo použito programů Amavisd-new, ClamAV a SpamAssassin. Po přijetí zprávy poštovním servrem je předána do Amavisu, který slouží jako externí filtr zpráv. Ten má za úkol předávát poštu pro antivirovou kontrolu programem ClamAV a antispamovou kontrolu SpamAssassinem.

ClamAV si pravidelně aktualizuje svoji virovou databázi a je provozovaný v systému jako démon. SpamAssassin využívá své vlastní filtry spolu s přídavnými moduly pro kontrolu pošty. Mezi přídavné moduly patří databáze kontrolních součtů Vipul's Razor, Pyzor a DCC. Z důvodu možnosti ověřovat obrázkové zprávy byl použit modul Fuzzy OCR pro převádění informací z obrázků do textové podoby. SpamAssassin také testuje odesílatele na existenci v černých seznamech. K tomuto účelu jsou použity SpamCop, DSBL, Spamhaus a NJABL. Po provedené kontrole Amavis rozhodne zda-li správu předá cílovému uživateli nebo ji uloží do karantény.

Konfigurační nastavení Amavisu jsou uložena v databázi MySQL ve formě bank politik. Vytvořené politiky je možné přiřazovat uživatelům aniž bychom museli restartovat běžící služby, vše se provede za plného provozu bez přerušení. Uživatelé si mohou vytvářet vlastní seznamy oblíbených nebo zakázaných odesílatelů. Zprávy, které neprojdou kontrolami skončí v karanténě, pokud by se jednalo o řádnou zprávu je možnost ji z karantény vytáhnout přímo do uživatelovi schránky. Nad celou karanténou jsou provedeny statistiky uvíznutých zpráv zpracované do grafů.

Celé uživatelské rozhraní Amavisu je provedeno jako webové aplikace vytvořené pomocí technologií HTML, CSS, PHP a MySQL. Přihlášení je možné vložením přihlašovacích údajů k poštovním schránkám. Rozhraní je dostupné na adrese http://titan.jirifrantisek.cz. Příloha obsahuje stručný návod pro ovládání aplikace. Samotný přínos pro autora bylo seznámení a proniknutí do technologií pro tvorbu webových aplikací, s kterými začal pracovat až na této práci.

Přidaná hodnota vytvořeného řešení je v možnosti podrobného nastavování antispamové ochrany uživatelem, což není možné u většiny dostupných poštovních služeb (Gmail, Seznam, atd.)

LITERATURA

[1] ADÁMEK, M. *Spam: jak nepřivolávat, nepřijímat a nerozesílat nevyžádanou poštu.* 1.vyd. Praha: Grada, 2009. 468 s. ISBN:978-80-247-2638-0

[2] BORONCZYK, T. - NARAMORE, E. - GERNER, J. - SCOUARNEC, Y. - STOLZ, J. - GLASS, M. *PHP 6, MySQL, Apache: Vytváříme webové aplikace* 1. vyd. Praha: Computer Press, 2009. 817 s. ISBN 978-80-251-2767-4.

[3] DENT, K. D. *Postfix: kompletní průvodce.* 1. vyd. Praha: Grada, 2005. 252 s. ISBN 80-247-1029-3.

[4] HILDEBRANDT, R. - KOETTER, P. *Postfix: Provozujeme poštovní server v Linuxu.* 1.vyd. Brno: Computer Press, 2006. 432 s. ISBN 80-251-1020-6.

[5] OPPEL, A. *SQL bez předchozích znalostí* 1. vyd. Praha: Computer Press, 2008. 241 s. ISBN 978-80-251-1707-1.

[6] SATRAPA, P. *Perl pro zelenáče* 1. vyd. Praha: Neocortex, 2001. 225 s. ISBN 80-86330-02-8.

[7] SCHAFER, S. *HTML, XHTML a CSS* 4. vyd. Praha: Grada Publishing, 2009. 649 s. ISBN 978-80-247-2850-6.

[8] SCHNEIDER, R. *MySQL: Oficiální průvodce tvorbou, správou a ladění databází* 1. vyd. Praha: Grada Publishing, 2006. 372 s. ISBN 80-247-1516-3.

[9] SYMANTEC, Co. *Symantec Internet Security Threat Report - Trends for 2010* 2011, poslední aktualizace březen 2011 [cit. 14.4.2011]

[10] ŠKULTÉTY, R. *JavaScript: Programujeme internetové aplikace.* 2.vyd. Brno: Computer Press, 2004. 224 s. ISBN:80-251-0144-4

[11] WOLFE, P. - SCOTT, C. - ERWIN, M. W. *Antispam: Metody, nástroje a utility pro ochranu před spamem.* 1.vyd. Brno: Computer Press, 2004. 376 s. ISBN 80-251-0479-6.

[12] Bug #517156 *Debian bug tracking system* 2009, [cit. 10.12.2010] <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=517156>.

[13] Courier Installation *Courier* [cit. 7.12.2010] <http://www.courier-mta.org/install.html>.

[14] GIBELLI, L. *ClamAV Wiki* 2010, poslední aktualizace listopad 2010 [cit. 11.12.2010]. Dostupné z URL: <http://wiki.clamav.net/Main/WebHome>.

[15] CRISPIN, M. *Internet Message Access Protocol - Version 4rev1.* 2003, poslední aktualizace březen 2003 [cit. 21.11.2010]. Dostupné z URL: <http://tools.ietf.org/html/rfc3501>.

[16] HOLLER, Ch. - VALDES, J. *Fuzzy OCR Wiki* 2007, poslední aktualizace červen 2007 [cit. 13.12.2010]. Dostupné z URL: <http://fuzzyocr.ownhero.net/wiki/WhatisFuzzyOcr>.

[17] KLENSIN, J. *Simple Mail Transfer Protocol.* 2008, poslední aktualizace říjen 2008 [cit. 2.11.2010]. Dostupné z URL: <http://tools.ietf.org/html/rfc5321>.

[18] MARTINEC, M. *Amavis general features* 2011, [cit. 14.3.2011]. Dostupné z URL: <http://www.amavis.org/#features-general>.

[19] MARTINEC, M. *Amavis: Lookups* 2005, [cit. 14.4.2011]. V Debianu dostupné z URL: </usr/share/doc/amavisd-new/README.lookups>.

[20] MARTINEC, M. *Amavis: Using SQL For Lookups, Reporting and Quarantine* 2009, [cit. 2.5.2011]. V Debianu dostupné z URL: </usr/share/doc/amavisd-new/README.sql-mysql>.

[21] MEHNLE, J. *SPF Record Syntax* 2008, poslední aktualizace červen 2008 [cit. 5.3.2011]. Dostupné z URL: <http://www.openspf.org/SPF-Record-Syntax>.

[22] MOCKAPETRIS, P. *Domain Names - Implementation and Specification.* 1987, poslední aktualizace listopad 1987 [cit. 15.11.2010]. Dostupné z URL: <http://tools.ietf.org/html/rfc1035>.

[23] MYERS, J. - MELLON, C. - ROSE, M. *Post Office Protocol - Version 3.* 1996, poslední aktualizace květen 1996 [cit. 19.11.2010]. Dostupné z URL: <http://tools.ietf.org/html/rfc1939>.

[24] MySQL 5.1 Reference manual *MySQL: Adding User Accounts.* 2010, [cit. 5.12.2010]. Dostupné z URL: <http://dev.mysql.com/doc/refman/5.1/en/adding-users.html>.

[25] NETCRAFT, Ltd. *Netcraft Web Server Survey, March 2011* 2011, poslední aktualizace duben 2011 [cit. 28.4.2011]. Dostupné z URL: <http://news.netcraft.com/archives/2011/03/09/march-2011-web-server-survey.html>.

[26] PARK, E. *Rustock Takedown's Effect on Global Spam Volume* 2011, poslední aktualizace březen 2011 [cit. 26.4.2011]. Dostupné z URL:

<http://www.symantec.com/connect/blogs/rustock-takedown-s-effect-global-spam-volume>.

[27] PATRIDGE, C. *Mail Routing and the Domain System.* 1986, poslední aktualizace leden 1986 [cit. 13.11.2010]. Dostupné z URL: <http://tools.ietf.org/html/rfc974>.

[28] Pyzor Wiki *Pyzor Wiki*225 2009, poslední aktualizace červenec 2009 [cit. 13.12.2010]. Dostupné z URL: <http://sourceforge.net/apps/trac/pyzor/wiki/About>.

[29] SCHRYVER, V. *Distributed Checksum Clearinghouse* 2010, poslední aktualizace listopad 2010 [cit. 13.12.2010]. Dostupné z URL: <http://www.rhyolite.com/dcc/>.

[30] The Public Linux Archive *Linux-PAM.* 2008, [cit. 6.12.2010]. Dostupné z URL: <http://www.kernel.org/pub/linux/libs/pam/>.

[31] TIMME, F. *Virtual Users And Domains With Postfix* 2009, [cit. 17.10.2010]. Dostupné z URL: <http://howtoforge.com/virtual-users-domains-postfix>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

Amavis A Mail Virus Scanner

CPAN Comprehensive Perl Archive Network

DKIM DomainKeys Identified Mail

DNSBL Domain Name System Black List

DNS Domain Name System

DNS TXT Domain Name System Text Record

DSN Delivery status notification

IMAP Internet Mail Application Protocol

ISP Internet Service Provider

LMTP Local Mail Transfer Protocol

MUA poštovní uživatelský agent – Message User Agent

MTA poštovní přenosový agent – Message Transfer Agent

MDA poštovní doručovací agent – Message Delivery Agent

NJABL Not Just Another Bogus List

OCR Optical Character Recognition

PAM Pluggable Authentication Modules

PGP Pretty Good Privacy

PHP PHP: Hypertext Preprocessor

POP Post Office Protocol

SASL Simple Authentication and Security Layer

SMTP Simple Mail Transfer Protocol

SPF Sender Policy Framework

SSL Secure Sockets Layer

UBE Unsolicited Bulk Email

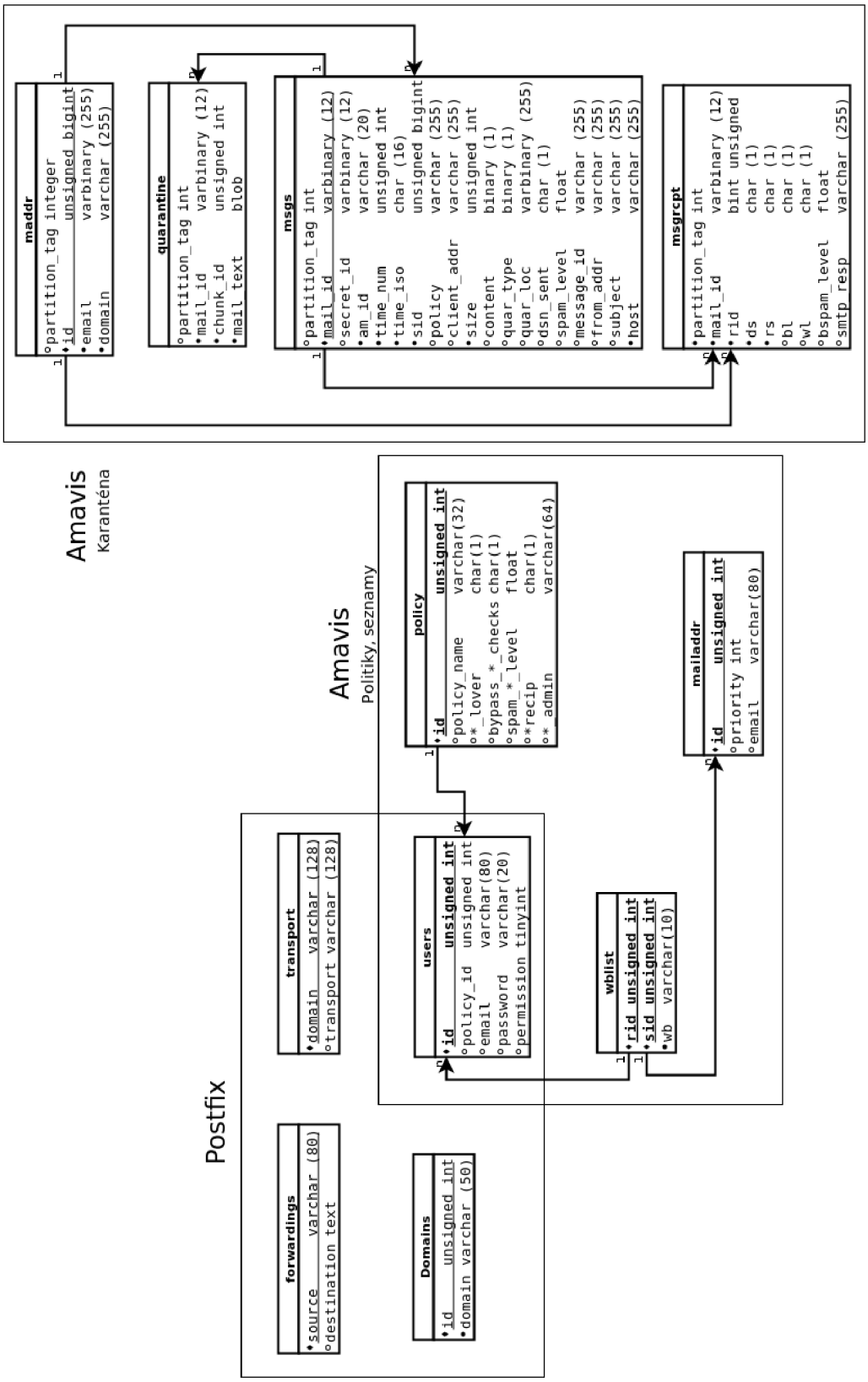
TLS Transport Layer Security

UCE Unsolicited Commercial Email

UZ uložiště zpráv

SEZNAM PŘÍLOH

A STRUKTURA DATABÁZE



B NÁVOD K APLIKACI

Aplikace je dostupná na webové stránce http://titan.jirifrantisek.cz. Testovací účet pro vyzkoušení má přihlašovací jméno test@jirifrantisek.cz a heslo je tvořeno čísly a znaky 'devět pět bphumeXj', číslovky se musí přepsat do numerické podoby. Po přihlášení máme na výber ze čtyř možností.

Politiky

Záložka „Politiky" slouží pro přidělování vytvořených politik uživatelům. Změnu provedeme vybráním politiky z roletky na řádku uživatele. Jakmile ukončíme výběr, tak stisknutím tlačítka „Uložit" se provedou změny v nastavení.

Nad seznamem uživatelů je funkční menu, kde je možné vybrat politiku pro editování nebo vytvořit novou. Pokud chceme změnit politiku stačí ji vybrat v roletce a zmáčknou tlačítko „Změnit". Pro vytvoření nové politiky je nutné vyplnit název politiky a zmáčknout tlačítko „Vytvořit". Políčko s názvem politiky nesmí zůstat prázdné, to vede k návratu do původního formuláře. Pokud bude vybrán existující název tak politika nepůjde uložit.

V editačním a vytvářecím formuláři politik si může uživatel přizpůsobit nastavení dle libosti. Výběrové roletky můžou mít hodnoty „null", „Ano", „Ne". Standardně je přednastavena hodnota „null", která indikuje nenastavení položky a její hodnota bude převzata ze základního nastavení. Textová políčka mohou obsahovat čísla (volby končící slovem „hranice"), e-mailové adresy a textové řetězce.

Po nadefinování politiky ji uložíme stisknutím tlačítka „Uložit". Pokud jsme vybrali v předchozí obrazovce editaci existující politiky tak tlačítkem „Smazat" můžeme vybranou politiku odstranit. Nesmí být nastavena u žádného uživatele!

B/W seznamy

K nastavování bílých a černých seznamů se dostane kliknutím na „B/W seznamy". Po vypsání nadefionových seznamů je můžeme smazat začkrtnutím volby na začátku řádku a kliknutím na tlačítko smazat vybrané. K vytvoření nového seznamu slouží tlačítko „Vytvořit". Po jeho stisknutí se přesuneme na formulář tvorby záznamu.

Z roletky vybereme místního příjemce. Do odesílatele můžeme vložit e-mailovou adresu nebo doménovou část adresy začínající zavináčem. Políčko hodnota slouží pro nastavení černého záznamu (vložíme písmeno B), bílého seznamu (vložíme písmeno W) nebo číslo, které se bude přičítat k výslednému ohodnocení zprávy v SpamAssassinu.

Karanténa

V záložce „Karanténa“ jsou zobrazeny zprávy uložené v karanténě. Uváděné údaje jsou datum přijetí zprávy, příjece, odesílatel, důvod uložení zprávy do karantény a předmět zprávy. Zaškrtnutím políčka na začátku řádku vybereme zprávy a stisknutím tlačítka „Obnovit“ provedeme jejich odeslání adresátům bez provedení kontroly.

Reporty

Na záložce „Reporty“ můžeme sledovat statistiky zpracovaných zpráv poštovním systémem. Na prvním grafu je zobrazen poměr přijatých regulérních a nevyžádaných zpráv. Další graf zobrazuje počty a druhy závadnosti zpráv uložených v karanténě.

Obě statistiky jsou dostupné pro všechny přijaté zprávy, zprávy doručené v posledních čtyrech týdnech a v posledním týdnu. Požadovanou statistiku je možné vybrat v rozevíracím menu pod záložkou.

Po provední potřebných změn je vhodné se odhlásit tlačítkem umístěným na pravém panelu. Tento panel také obsahuje odkaz na webové prezentace používaných prvků v boji proti nevyžádané poště.

C ELEKTRONICKÁ PŘÍLOHA

Obsah přiloženého CD:

- Diplomová práce ve formátu pdf.
- Zdrojové kódy webové aplikace.
- Konfigurační soubory Postfixu.
- Konfigurační soubory Amavisu.
- Konfigurační soubory SpamAssassinu.
- Struktura databáze MySQL.