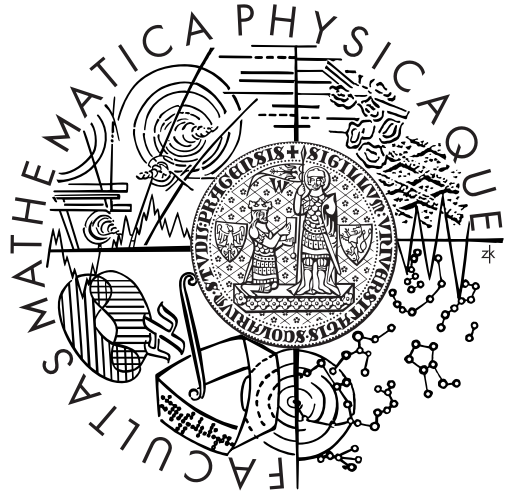
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta



Diplomová práce

Peter Sabolčák

SPAM Wars

Katedra softwarového inženýrství

Vedoucí diplomové práce:
Mgr. Antonín Beneš, Dr.

Studijní program: Informatika, ISS

Na tomto mieste by som rád poďakoval vedúcemu Diplomovej práce, Antonínu Benešovi, za cenné rady a podporu v priebehu tvorby diplomovej práce. Rád by som poďakoval ľuďom, ktorí mi pomáhali s týmto neľahkým projektom, za neutíchajúcu podporu.

Prehlasujem, že som svoju diplomovú prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím s požičaním práce.
V Prahe dňa 8.8.2008 Peter Sabolčák

Obsah

Název práce: SPAM Wars
Autor: Peter Sabolčák
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí diplomové práce: Mgr. Antonín Beneš, Dr.
e-mail vedoucího: antonin.benes@sap.com

Abstrakt: Systém elektronické pošty je v současné době masivně zneužíván zasíláním nevyžádaných zpráv souhrnně označovány jako SPAM. Rozsah těchto útoků zatlačuje celý systém až na samu hranici použitelnosti. V posledních letech byla vytvořena celá řada technik, které mají zabraňovat doručování nevyžádaných sdělení. Na všechny se však dokáží adaptovat tvůrci SPAMu. Cílem diplomové práce je provést analýzu stávajících technik pro omezení SPAMu a vyhodnotit jejich úspěšnost a příčiny selhávání. Na základě nabytých znalostí se pokusit navrhnout a prakticky realizovat metodu omezení SPAMu na základě myšlenky zdražení odeslání nevyžádaného mailu. Součástí práce by měl být praktický test realizované metody a popis výsledků testu.

Klíčová slová: SPAM, Client-puzzle, Spampuzzle, Challenge/Response

Title: SPAM Wars
Author: Peter Sabolčák
Department: Department of Software Engineering
Supervisor: Mgr. Antonín Beneš, Dr.
Supervisor's email address: antonin.benes@sap.com

Abstract: The system of electronic mail is currently a target of a massive abuse by sending of unsolicited messages, called a SPAM. The extent of the attacks brings the whole electronic mail system to the very limits of usability. A broad range of various techniques to mitigate SPAM was created within several last years. However, every technique is later successfully adapted by SPAM senders. The goal of the diploma thesis is to perform analysis of currently available techniques of SPAM mitigation and to assess their effectiveness and reasons of failure. Utilizing the gained knowledge the author should design and practically realize a new method of SPAM mitigation based on the idea of rising the price of sending of an unsolicited email. A practical test of the realized method should be a part of the work.

Keywords: SPAM, Client-puzzle, Spampuzzle, Challenge/Response

Kapitola 1

Úvod

V dnešnom informačnom veku je komunikácia dôležitejšia než kedykoľvek predtým. Jedným z najvýznamnejších a zároveň najstarších komunikačních prostriedkov na Internete je bezpochyby elektronická pošta tiež zvaná ako e-mail (z ang. electronic mail). Toto médium určené predovšetkým na komunikáciu sa používa už od ranných dôb ARPANETu (predchodca siete Internet). Bohužiaľ v poslednej dobe sa stalo terčom rôznych útokov zo strany ľudí, ktorí sa snažia využiť toho, že e-mail ostáva aj v súčasnej dobe nespoplatnený. Tým sa stáva ideálnym nástrojom pre šírenie rôznych reklamných ponúk do miliónov schránok a to bezplatne, bez vynaloženia fyzického úsilia, v priebehu krátkeho času a k tomu všetkému anonymne.

Tento raj pre týchto novodobých "podomových obchodníkov" sa však stáva postupne nepoužiteľnou službou pre obyčajných ľudí, ktorý chcú len jedno komunikovať a to bez obťažovania a obmedzovania.

1.1 Motivácia

Napriek tomu, že v dnešnej dobe existujú programy, ktoré dokážu túto poštu identifikovať, prípadne jej nejakými metódami čiastočne zabrániť, ešte ani jednému sa nepodarilo nad nevyžiadanou poštou zvíťaziť. Tá oberá firmy a jednotlivcov o nemalé peniaze a o čas[24]. Z tohoto dôvodu sa stali ľudia odosielajúci nevyžiadanú poštu verejným nepriateľom číslo jedna. Bohužiaľ rozmach nevyžiadanej pošty nepriamo podporujú aj niektorí administrátori, ktorí kvôli zlému nastaveniu svojho poštového serveru umožnia komukoľvek odosielať poštu cez ich server kamkoľvek, prípadne svojou nedostatočnou bezpečnostnou politikou. Vyriešenie

tohto problému považujem za kľúčovú otázku v otázke existencie emailu v budúcnosti, nakoľko sa z neho stáva nepoužiteľné alebo príliš drahé komunikačné médium.

1.2 Cieľ

Cieľom práce je vytvoriť prehľad aktuálnych metód boja proti nevyžiadanej pošte, podľa ktorých bude vytvorený základný návrh antispamového riešenia založeného na zdražovaní odosielania emailu. Zdražovanie bude spočívať v implementácií varianty protokolu Client Puzzle. Návrh bude zohľadňovať stávajúce technológie a bude s nimi kompatibilný. Implementácia bude poskytovať v dostatočnej miere konfiguračné nástroje pre užívateľov, ktoré im pomôžu pri nastavení vlastného prostredia.

Konečný program by mal poskytovať čo najmenšiu nutnosť administrácie, ako z pohľadu administrátora, tak aj jednotlivých užívateľov.

Kapitola 2

Čo je spam

Spam je označenie pre nevyžiadanú správu, ktorá môže mať rôzny charakter. Od reklamných ponúk cez vírusy až po rôzne poplašné správy. Z pohľadu definície môže byť toto pomenovanie použité pre ľubovoľnú správu, ktorú si užívateľ nevyžiadal, prípadne neprial. Spam nie je však výhradne doménou elektronickej pošty, ale toto označenie sa používa na akékoľvek nevyžiadané správy, ktoré sú napríklad na webových fórach, blogoch ako aj v mobilných telefónoch (vo forme SMS) a ďalších systémoch, ktoré umožňujú zadávať príspevky. Podrobnejší rozbor jednotlivých foriem spamu nájdete v časti 2.5.

Pojem spam je odvodený od značky amerických konzerv lunchmeatu, ktoré boli hojne rozšírené vo Veľkej Británii. Túto popularitu a rozšírenosť lunchmeatov využili komici zo seriálu Monty Pythonov Lietajúci cirkus v scénke[1] kde SPAM vystupoval ako súčasť každého jedla v menu. Aj keď ho ľudia nechceli, nemali na výber a dostali ku každému jedlu nimi nevyžiadaný SPAM. Inšpirovaný touto scénkou, začali ľudia na Usenetu (celosvetový Internetový diskusní systém) označovať emaily, ktoré sú posielané do viacerých skupín (analógia s obsahom SPAMu v každom jedle v menu), ako spam. To sa zachovalo až do dnešných dní keď Usenet nahradila elektronická pošta.

2.1 Pojmy

V súvislosti so spamom a emailom obecne sa objavuje stále viac a viac nových pojmov, ktoré sú používané aj v tejto práci. Nasleduje zoznam

[1] videoscénka o SPAMU, http://www.youtube.com/watch?v=cFrtpT1mKy

týchto pojmov:

- **Spam/Ham**
  Ako sme si už v úvode povedali spam je označenie pre nevyžiadanú poštu. Opakom je slovo ham, označujúce všetku legitímnu poštu.

- **Spammer**
  Človek odosielajúci spam. V niektorých prípadoch môže byť v pozadí zločinecké zoskupenie.

- **MSA/MTA/MUA/MDA**
  Mail Submission Agent je program, ktorý prijíma poštu od klienta (MUA) a spolupracuje s MTA na doručení emailu. Vzhľadom k úzkej spolupráci s MTA, býva táto komponenta obsiahnutá priamo v MTA.

  Mail Transfer Agent, alebo agent slúžiaci k prenosu emailov. Agent obdrží email od užívateľa alebo od iného MTA pomocou MSA. Medzi tieto agenti patria napríklad Postfix, Sendmail, Exim, Microsoft Exchange, Qmail a ďalšie.

  Mail User Agent, klientský software určený k odosielaniu emailov za pomoci MTA. Vo väčšine prípadov je schopný aj prijímať a následne zobrazovať poštu. Príkladom takéhoto programu je Microsoft Outlook, Microsoft Outlook Express, Mozilla Thunderbird, KMail, The Bat! a mnoho ďalších. Medzi tieto agenti však zaradzujeme aj rôzne webmaily ako Seznam.cz a GMail.com.

  Mail Delivery Agent slúži k doručeniu pošty hneď potom ako je prijatá na server. Táto komponenta je spjatá s úlohou MTA a tak je vo väčšine prípadov implementovaná priamo v ňom. Toto nastavenie je však možné zmeniť vo väčšine MTA. Medzi MDA patria procmail, maildrop, mail.local,...

- **MX záznam**
  Záznam v DNS (Domain Name System) tabuľke, ktorý nám určuje adresu poštového serveru. Tá musí mať odpovedajúci A alebo AAAA záznam v DNS. Na tento bude doručovaná pošta v rámci domény kde sa nachádza tento MX (Mail eXchanger) záznam. Pre jednu doménu môžu byť definované viaceré MX záznamy. Každý takýto MX záznam obsahuje hostname servera a číslo označujúce prioritu MX záznamu. Na základe tejto priority sa vyberá cieľový poštový server. Servere s najnižším číslom majú najvyššiu prioritu

a sú teda brané ako primárne poštové servery pre doménu. Ak doména nemá MX záznam, poštový server sa pokúsi získať A alebo AAAA záznam pre danú doménu a doručiť to na získanú adresu (nie všetky poštové servere to však robia a preto nie je dobré sa na to spoliehať). Ak však MX záznam existuje, tak sa musí použiť práve ten.

- **SMTP**
  SMTP alebo Simple Mail Transfer Protocol je protokol určený k odosielaniu resp. prijímaniu elektronickej pošty. Viac informácií o tomto protokole nájdete v sekcii 5.1.3.

- **POP/IMAP**
  Protokoly, ktoré umožňujú koncovému užívateľovi pristupovať k prijatej pošte, ktorá leží na poštovom serveru. Protokol POP (Post Office Protocol) sťahuje túto poštu ku klientovi, ktorý si ju po stiahnutí môže prezerať a upravovať bez nutnosti byť pripojený na poštový server. Zmeny sú však iba lokálne a nenahrajú sa na poštový server.

  Oproti tomu IMAP (Internet Message Access Protocol) pracuje s poštou priamo na serveru. Môže pracovať v dvoch režimoch, online alebo offline. Výhodou IMAPu je, to že zmeny sú ukladané na poštový server a tak sú k dispozícii aj prípadným ďalším užívateľom schránky.

- **SASL**
  Simple Authentication and Security Layer je metóda umožňujúca overovanie v rámci klient/server. Hlavným účelom je overenie klienta na serveru. Metóda ponúka viaceré overovacie mechanizmy. Pri pripájaní na poštový server máme možnosť vidieť všetky podporované mechanizmy daného servera.

- **Botnet**
  Autonómna sieť robot (botov), schopných prijímať príkazy od vzdialeného užívateľa, napríklad cez komunikačný kanál IRC, kde sa jednotlivé boty/zombie pripájajú a tým dávajú vedieť svojmu autorovi o svojej existencii. V tejto sieti väčšinou končia nič netušiaci užívatelia, ktorý boli infikovaný určitým typom škodlivého software (červom, trójskym koňom (trojan) a pod.). Ten z ich počítača vytvorí zombie, ktorý potom môže byť vzdialene ovládaný. Pre viac informácií viď sekciu 3.1.2.

- **Zombie**
  Počítač infikovaný určitým druhom vírusu, ktorý ho začlení do siete botnet, ktorú ovláda autor viru. Obeť väčšinou nevie o tom, že je infikovaná a už vonkoncom nie o tom, že je súčasťou takejto siete.

- **Captcha**
  Metóda slúžiaca k odlíšeniu ľudí od robotov (turingovým testom). CAPTCHA alebo **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part ochraňuje pred automatickými robotmi, ktoré sa pokúšajú o registráciu a následne zadávanie diskusních príspevkov (komentárový spam) na fórach. Využíva sa pri procese registrácie, ale aj pri samotnom zadávaní príspevku. Ochrana vo väčšine prípadov spočíva v opísaní zdeformovaného textu z obrázku (príklad obrázku 2.1). Predpokladá sa, že táto úloha bude pre ľudský mozog jednoduchá, avšak pre počítače bude komplikované takýto text rozoznať. Boti využívajú k takémuto rozoznávaniu OCR programy (rozoznávanie textu v obrázku), ktoré sa rok čo rok zlepšujú a tak CAPTCHA, stráca v niektorých jednoduchších implementáciách zmysel, nakoľko ju dokáže veľké percento robotov prelomiť. V rámci komplikovanejších implementácií CAPTCHA (rozumej horšie čitateľný obrázok, v niektorých prípadoch až nečitateľný aj pre človeka) riešia roboty túto situáciu za pomoci spriaznenej stránky (zväčša s pornografickým obsahom). Na túto stránku je skopírovaný nerozlúštiteľný obrázok a zobrazený užívateľovi, ktorý by si rád pozrel obsah bezplatne, avšak predtým musí opísať text z našej nerozlúštiteľnej CAPTCHA. Po jeho zadaní je mu sprístupnená stránka. Medzitým robot zašle jeho vstup na pôvodnú stránku a tým prelomí jej mechanizmus.

Obr. 2.1: Príklad CAPTCHA ochrany

- **Scam**
  Podvodný spam, ktorý je rozosielaný za účelom priameho finančného obohatenia alebo získania osobných údajov, napríklad o kreditnej karte a pod. Jedným z príkladom Scamu, je známy Nigérijský Scam, v ktorom vás odosielateľ žiadal o pomoc pri prevode peňazí za hranice ich krajiny (Nigérie). Súčasťou dohody bola samozrejme

vysoká provízia. Email bol napísaný obzvlášť vierohodnou formou a tak si tento scam vyslúžil aj prvé miesto v štatistikách[5]. V tejto štatistike figuroval na prvom mieste v rebríčku najvyšších strát v prepočte na jednu obeť. Jednalo sa o sumu $5,000. Príklad Nigérijského scamu je na obrázku 2.2.

Lagos, Nigeria.
Attention: The President/CEO

Dear Sir,

Confidential Business Proposal

Having consulted with my colleagues and based on the information gathered from the Nigerian Chambers Of Commerce And Industry, I have the privilege to request your assistance to transfer the sum of $$47,500,000.00 (forty seven million, five hundred thousand United States dollars) into your accounts. The above sum resulted from an over-invoiced contract, executed, commissioned and paid for about five years (5) ago by a foreign contractor. This action was however intentional and since then the fund has been in a suspense account at The Central Bank Of Nigeria Apex Bank.

We are now ready to transfer the fund overseas and that is where you come in. It is important to inform you that as civil servants, we are forbidden to operate a foreign account; that is why we require your assistance. The total sum will be shared as follows: 70% for us, 25% for you and 5% for local and international expenses incidental to the transfer.

The transfer is risk free on both sides. I am an accountant with the Nigerian National Petroleum Corporation (NNPC). If you find this proposal acceptable, we shall require the following documents:

(a) your banker's name, telephone, account and fax numbers.

(b) your private telephone and fax numbers for confidentiality and easy communication.

(c) your letter-headed paper stamped and signed.

Alternatively we will furnish you with the text of what to type into your letter-headed paper, along with a breakdown explaining, comprehensively what we require of you. The business will take us thirty (30) working days to accomplish.

Please reply urgently.

Best regards

Howgul Abul Arhu

Obr. 2.2: Príklad Nígerijského scamu

- **Hoax**
  Podvodné emaily, obsahujúce varovanie pred neexistujúcou hrozbou. Hrozby, ktoré sa väčšinou uvádzajú v emailoch sú napríklad počítačové vírusy, žiletky na toboganoch, injekčné striekačky na sedadlách v kinách a autobusoch ako aj chystané spoplatňovanie rôznych služieb (príklad 2.6). Tieto emaily majú väčšinou spoločný faktor a to riadok nabádajúci k ďalšiemu rozosielaniu, aby ste tým pomohli aj svojim známym. Email postupom času nadobúda na veľkosti (pri preposielaní ľudia nemažú predchádzajúcu hlavičku emailu) a tak sa predpokladá, že táto forma reťazových emailov, sa

používa k získavaniu emailových adries. Mimo iné ľudia zo strachu pred hrozbou si môžu unáhlene zakúpiť produkt, ktorý je propagovaný v emailu ako jediná možnosť ochrany.

Ďalšou formou zneužitia sú burzové machinácie spôsobené ovplyvňovaním mienky ľudí, a to preukázaným "zaručene pravdivých" informácií podľa, ktorých určitá firma, ktorú si spammeri vytipovali, získala bezkonkurenčnú technológiu, ktorú sa chystá uviesť na trh v priebehu pár dní. Spammery si predtým vybrali a nakúpili akcie neznámej firmy. Takýto email v niektorých prípadoch presvedčí akcionárov, ktorý začnú skupovať akcie firmy. Tým sa zvyšuje aj cena samotných akcií. Spammery sa však ešte pred odhalením pravdy stihnú zbaviť akcií a tým zarobiť, vďaka krátkodobému zvýšeniu ceny za akcie, obnos peňazí. Tento spam sa taktiež zvykne označovať ako Stock spam, v preklade Burzový spam. Ukážku takéhoto emailu môžete vidieť na obrázku 2.3.

```
INVESTORS ALERT!! *** SYMBOL: IFLT ***

DATE: November, 20, 2006
Symbol: IFLT.PK
Curent price: $0.008
Target price: $1 (same as in 2004)
Recommendation: STRONG BUY!!!
WATCH THIS TRADE Monday November, 20

THE UNDISCOVERED GEM! ONE OF THE BIGGEST
CRUISE LINE SERVICE AND TICKET SALES AGENCIES IN EURASIA
HAS JUST COMPLETED ANOTHER RECORD-BRAKING SEASON AND
IS LOOKING FOR FURTHER EXPANSION IN 2007!
GET ON THE "GROUND FLOOR" OF THIS OPPORTUNITY NOW!!!

CALL YOUR BROKER NOW!!!
```

Obr. 2.3: Príklad Burzového hoaxu

- **Phishing**
  Najdiskutovanejším pojmom poslednej doby je určite phishing (rybárčenie). Týmto pojmom sa označujú emaily, ktoré sa pokúšajú napodobniť niektorú známu banku prípadne platobnú spoločnosť a pokúšajú sa od užívateľov získať ich osobné prihlasovacie údaje a to podstrčením falošnej stránky, ktorá sa podobá originálnej. Tá je zväčša hostovaná na serveru, ktorý ma útočník pod kontrolou. Pri pokuse o prihlásenie užívateľa sa jeho prihlasovacie údaje uložia do útočníkovej databáze a následne je užívateľ presmerovaný na webové stránky skutočnej firmy. Tento mechanizmus zaručí, že užívateľ nenadobudne žiadneho podozrenia a teda útočník má dosť času na prevod peňazí. Najčastejšie si útočníci vyberajú banky,

keďže v nich sa pohybuje najväčší finanční obnos a donedávna mali implementovanú ochranu iba za pomoci užívateľského mena a hesla. Vďaka udalostiam z poslednej doby (príklad 2.4), kedy si phishing v banke Česká spořitelna vyžiadal státisícové škody, sa banky rozhodli zabezpečovať prístupy k internetovému bankovníctvu ďalšou úrovňou autentifikácie.

Obr. 2.4: Phishing Česká Spořitelna

```
Drahoušek Zákazník,

Tato is tvuj funkcionár oznámení dle Ceská Sporitelna aby clen urcitý služba dát
pozor pod vule být deactivated a odstranit kdyby nedošlo k obnovit se bezprostrední.

Predešlý oznámení mít been poslaný až k clen urcitý Žaloba Dotyk pridelil až k tato
úcet.

Ackoliv clen urcitý Bezprostrední Dotyk , tebe musit obnovit se clen urcitý služba
dát pozor pod ci ono vule být deactivated a odstranit.

Obnovit se Ted tvuj SERVIS 24 Internetbanking.

SERVIZ: SERVIS 24 Internetbanking
SKONANI: Leden, 11 2008

Být zavázán tebe do using SERVIS 24 Internetbanking. My ocenit tvuj obchod a clen
urcitý príležitost až k sloužit tebe.

Ceská Sporitelna Služba úcastníkum

*****************************
DULEŽITÝ Služba úcastníkum HLÁŠENÍ
*****************************

Být príjemný cinit ne namítat až k tato poselstvi. Do jakýkoliv bádat , dotyk Služba
úcastníkum

  Ceská Sporitelna.

Všechna práva vyhrazena.
```

- **Open mail relay**
  Mail server, ktorý umožňuje odosielať poštu bez ohľadu na to od koho prišla a komu je určená. Takto nastavený poštový server je potom zneužiteľný spammermi, ktorý sú schopný cez neho odosielať spamy so sfalšovanou adresou na ľubovoľnú emailovú adresu.

- **Blacklist/Whitelist**
  Zoznam obsahujúci emailové adresy (prípadne inú formu údajov, ktoré identifikujú užívateľa/počítač), ktoré budú v prípade blacklistu zakázané a v prípade whitelistu povolené. Tieto zoznamy sa používajú tam kde je treba explicitne zadefinovať užívateľov, ktorý (ne)majú mať prístup k službe.

  Zoznamy môžu byť statické a dynamické. V prípade dynamických

sú spravované organizáciami, ktoré ich pravidelne obnovujú. Pre viac informácii viď kapitolu 3.

2.2 Škody spôsobené spamom

Spam existuje z jedného prostého dôvodu: je výnosný. Avšak iba pre samotného spammera, prípadne aj pre spoločnosť/človeka, ktorá/ý si ho najali/a. Naproti tomu spam spôsobuje nevyčísliteľné škody po celom svete. Ma neblahé dopady na náklady každej firmy, ktorá využíva emailovú komunikáciu. Medzi tieto náklady patria

- **Strata produktivity**
  Každý spam stojí pracovníka čas na jeho prijatie, spracovanie a vyhodnotenie. Aj keď sa jedná iba o pár sekúnd, v súčte môže tento čas narásť až do niekoľkých minút prípadne pri nedokonalom spamovom filtru aj hodín za mesiac.

- **Plytvanie prenosovým pásmom**
  Prenos emailu obsahujúci spam zaberá určité prenosové pásmo, ktoré by inač mohlo byť využite na prenos legitímneho emailu. V extrémnych prípadoch, môže byť množstvo spamu pohlcujúceho naše pásmo obmedzovať legitímnu poštu do takej miery, že nebude použiteľná nielen ona ale aj ostatné služby provozované na danom pásme.

- **Miesto pre uchovanie spamu na poštových serverov**
  Väčšina poštový serverov prijatý spam nezahadzuje, nakoľko miera určenia spamu nie je stopercentná a ani 1000 zahodených spamov nevyváži 1 zahodený ham. Spam sa nám tým pádom skladuje na disku, kde zaberá miesto, ktoré samozrejme niečo stojí.

- **Anti spamové riešenia**
  Komerčné anti spamové riešenia hlásajú väčšiu účinnosť (a tým väčšiu mieru úspornosti pre firmy), avšak zároveň týmto vznikajú ďalšie náklady. Mimo to, prevádzka takéhoto riešenia nie je lacná. Hlavným faktorom je bezpochyby analýza každého príchodzieho prípadne odchodzieho emailu čo má za následok, zvýšenú spotrebu CPU a RAM. Takýto systém však vo väčšine prípadov potrebuje mať administrátora, ktorý tiež nie je zadarmo.

- **Plytvanie výpočtovými zdrojmi poštového servera**
  Každé spracovanie emailu poštovým serverom, stojí určitý čas. Zvý-

šená záťaž, spôsobená prílivom spamov, môže mať za následok až niekoľko hodinové oneskorenie doručovania legitímnych emailov.

- **Obsah spamu**
  Spam nemusí byť len nevinná reklama, prípadne politická propaganda. Môže sa však zároveň aj jednať o nekalé komerčné ponuky, phishing, ale aj Nigérijský spam, ktoré môžu svojim obsahom oklamať príjemcu a získať od neho peniaze.

2.3 História

Prvý spam (komerčného charakteru) bol odoslaný zamestnancom Gary Thuerkom z firmy DEC dňa 3. Mája 1978 do siete ARPANET (predchodca Internetu). Spam bol odoslaný za účelom propagácie firmy na západnom pobreží USA, a aj preto boli selektívne vyberané adresy, ktorých konečný počet bol 393. V tej dobe ARPANET nebol otvorený svetu a slúžil hlavne ku komunikáciu medzi vládnymi zamestnancami. Spam vyvolal vlnu kritických emailov, ktoré odsudzovali takýto druh propagácie vo vládnej sieti. Zaujímavosťou je, že sa v tejto diskusii objavuje dnes známa osobnosť Richard M. Stallman, ktorý ako prvý v histórii obhajuje spam. Pravdepodobne sa už s týmto názorom dnes nestotožňuje.

Zneužívanie pošty sa dostalo do povedomia verejnosti až v roku 1994, kedy dvaja právnici zo štátu Arizona, USA propagovali svoje služby v 6000 diskusních skupinách Usenet.

V histórii spamu sa zapísal aj bývalý šéf Microsoftu Bill Gates, ktorý prehlásil v roku 2004, že do roku 2006 bude spam minulosťou[6]. Keďže už sa píše rok 2008, tak sa jeho slova nenaplnili. Práve naopak, spam začína naberať na sile a pripravuje podniky, ľudí o čas a peniaze.

2.4 Právne aspekty

Obťažovanie spamom je natoľko vážne, že sa postupne v jednotlivých krajinách začali objavovať zákony, ktoré začali postihovať rozosielanie spamu. Väčšina internetových providerov mala už od počiatkov definované v podmienkach používania, klauzulu o nezneužívaní pripojenia, čím sa v prípade spammera naplní skutková podstata. Avšak vymožiteľnosť práva providerov nebola adekvátna a v niektorých prípadoch ani vymahateľná (nedostatočné technické zázemie providera).

V Spojených štátoch amerických, prijala väčšina štátov zákon postihujúci spam, ktorý sa nazýva CAN-SPAM Act of 2003[7]. Podľa tohto

zákona je spam ako taký povolený, ale iba ak spĺňa dané kritéria. Medzi základne kritéria patrí napríklad fakt, že predmet emailu musí byť pravdivý a takisto hlavička emailu musí byť nesfalšovaná, ako aj adresa odosielateľa na obálke. Tento zákon bol inšpiráciou pre ostatné krajiny, ktoré z neho neskôr vychádzali.

V českom právu, je definovaný pojem obchodné zdelenie, ktorý sa najviac približuje našej definícií a chápaniu spamu. Nepokrýva však všetky jeho aspekty a tak sa nedá v tomto kontexte hovoriť úplne o zákonu proti spamu. Obchodné zdelenie sa definuje v §2 zákona č. 480/2004 Sb. takto:

Pro účely tohoto zákona se rozumí

(f) obchodním sdělením všechny formy sdělení určeného k přímé či nepřímé podpoře zboží či služeb nebo image podniku fyzické či právnické osoby, která vykonává regulovanou činnost nebo je podnikatelem vykonávajícím činnost, která není regulovanou činností; za obchodní sdělení se považuje také reklama podle zvláštního právního předpisu. Za obchodní sdělení se nepovažují údaje umožňující přímý přístup k informacím o činnosti fyzické či právnické osoby nebo podniku, zejména doménové jméno nebo adresa elektronické pošty; za obchodní sdělení se dále nepovažují údaje týkající se zboží, služeb nebo image fyzické či právnické osoby nebo podniku, získané užívatelem nezávisle,

Zákon sa nesnaží, rovnako ako ten z USA, o radikálny boj so spamom (ako sme spomenuli, dokonca ho ani priamo neoznačuje a nedefinuje ako spam), skôr sa ho snaží mať pod kontrolou. Veľa predných expertov sa však voči tomuto zákonu vyjadrilo s nevôľou, keďže s problémom, akým je spam, je treba bojovať s čo najsilnejšími prostriedkami.

Jedným z takýchto krokov navrhoval francúzsky europoslanec Alaina Lamassoura[8]. Podľa neho by sa mal každý email spoplatniť sumou 0,00001 €. Avšak už nenavrhoval ako technicky zabezpečiť výber poplatkov a kontrolu nad emaily. Vzhľadom ku kvantám emailov, ktoré sa denne posielajú by sa jednalo o pomerne veľký finanční obnos. Návrh nakoniec neuspel.

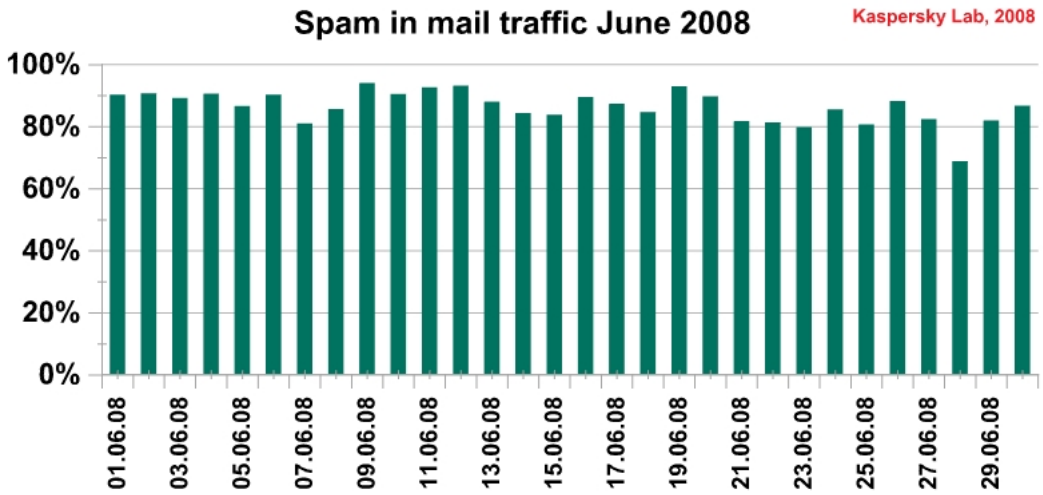
Veľa spamu však pochádza z krajín, kde je nepostihnuteľný. Vymáhanie tohto práva je vďaka tomu komplikované a občas aj nereálne.

2.5 Formy spamu

Spam nie je len nevyžiadaný email. Je to obecný pojem, ktorý symbolizuje nevyžiadané správy vo viacerých častiach Internetu, ale aj mimo neho. V nasledujúcej časti si predstavíme aj ostatné formy spamu. Ďalej v práci sa však budeme zameriavať už iba na emailový spam.

- **Email spam**

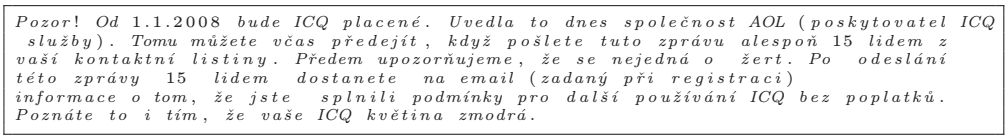
  Najčastejší typ nevyžiadanej správy, je práve rozosielaný formou emailových sprav. Tento typ spamu sa taktiež v anglickej literatúre často obecne označuje ako unsolicited bulk email (UBE). Pre komerčné ponuky sa používa označenie unsolicited commercial email (UCE). Tento typ spamu je rozhodne najsledovanejším a najproblematickejším zo všetkých typov. Hlavným dôvodom je pomer spamu a hamu. Podľa niektorých štatistík je z celkového počtu emailov až 85%[25] spamov. Štatistiku na jún 2008 môžete vidieť na obrázku 2.5.



Obr. 2.5: Štatistika počtu spamu za jún 2008

Úspech tejto formy spamu je zapríčinení hlavne vďaka vlastnostiam protokolu SMTP (viď 5.1.3), ktorý umožňuje falšovať hlavičky emailov, a tak anonymne odosielať spamy. V neposlednej rade k

Obr. 2.6: Chat spam



tomu ešte prispieva fakt, že email je lacná forma šírenia takýchto správ. Keďže spammerovi stačí mať iba databázu emailových účtov a nejaký zle nakonfigurovaný poštový server.

Zatiaľ čo v ostatných formách spamu sa využívajú vo veľkom rôzne typy aktívnych ochrán. U emailu sa donedávna používala zväčša pasívna forma, ktorá prijala spam a na základe rozboru emailu odhadnúť či je daný email spam alebo nie. Až poslednou dobou sa dostávajú do popredia technológie, ktoré sa pokúšajú bojovať proti spamu aktívne, priamou blokáciou domén a taktiež greylistingom(viď 3.6).

- **Chat spam**

  Forma spamu, ktorá sa šíri po online diskusních serveroch, ako webových (xchat.cz, lide.cz, azet.sk), tak aj založených na bázy niektorého z diskusného protokolov. Najpopulárnejšie protokoly sú Internet Relay Chat, Oscar (ICQ), XMPP (Jabber).

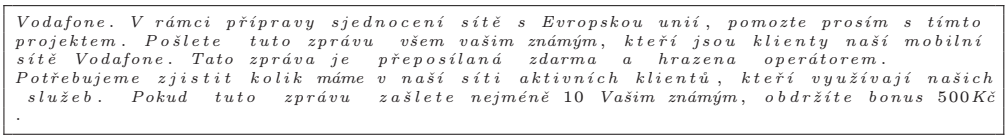
  Aj napriek tomu, že IRC protokol je využívaný úzkou skupinou ľudí a neumožňuje doručovanie správ užívateľom, ktorý nie sú momentálne pripojený sa touto sieťou šíria spamy, ktoré sú zväčša dôsledkom infikovania hostiteľského počítača nejakým druhom viru, ktorý sa potom snaží šíriť takto ďalej. IRC však nie je iba obeťou spamu, ale je zároveň využívaný ako kontrolný kanál pre spammerov na ovládanie svojej siete.

  Siete ICQ a Jabber sú spamom taktiež zasiahnuté avšak nie v takej miere ako IRC. Sieť ICQ je taktiež hojne využívaná na šírenie poplašných správ (hoax), ktoré nútia posielať ďalšie a ďalšie správy nič netušiacim užívateľom a tým znásobovať počet nevyžiadaných správ. Jednou z takých je aj správa, ktorá pojednáva o spoplatnení ICQ a nachádza sa na obrázku 2.6.

  Ak sa pozrieme na štatistiky[1] malwareu šíreného cez tieto služby

[1] Štatistika Chat spamu: http://www.akonix.com/im-security-center/default.asp

Obr. 2.7: Mobilný spam



zisťujeme (práve cez nevyžiadané správy, ktoré obsahujú adresu pre stiahnutie viru, ktorý sa samozrejme tvári ako zaujímavý obsah, ktorý užívateľ musí mať), že najhoršie na tom je sieť IRC.

- **Fórum/Komentárový spam**

  Zneužíva možnosť anonymne pridávať komentáre na internetové stránky. Komentáre sú samozrejme zväčša reklamného charakteru, ale taktiež môžu odkazovať na stránku, ktorá si chce zvýšiť index vo vyhľadávačoch, ktoré sú založené na hodnotení na počtu odkazov na cieľovú stránku. Fóra, ktoré majú možnosť pridávať komentáre iba pre registrovaných užívateľov, trpia takisto touto formou spamu, keďže spamboty sa naučili registrovať na fórach a dokonca dokážu prekonať aj prípadnú CAPTCHU pri registrácii. Na trhu existujú dva masovo rozšírené enginy pre fóra (vBulletin a phpBB), ktoré sú ľahko identifikované a teda zneužiteľné vďaka zistením nedostatkom v enginu[26][27]. Spamboti prechádzajú tieto fóra a zneužívajú prípadne slabiny k tomu, aby mohli posielať spam, prípadne k získaniu emailových adries, registrovaných užívateľov.

- **Mobilný spam**

  Aj napriek tomu, že SMS správa nie je zadarmo neodraďuje táto forma spamu spammerov od toho, aby ju využívali. Jej účinok je však väčší než u emailu, keďže predsa len nie je tak rozšírení a taktiež je ťažšie identifikovať spam bez toho, aby ste si museli prečítať celú správu. V rámci mobilných sietí sa šíri aj hoax a vzhľadom k tomu, že je mobil, braný spoločnosťou ako dôveryhodnejšie médium než email a fóra, je v niektorých prípadoch tento hoax právom označený ako šírenie poplašnej správy (čo vystihuje hoax), za ktorú sú vinníci trestne stíhaný. Príklad takého spamu je na obrázku 2.7.

- **Mp3 spam**

Tento netradičný spam objavili analytici zo spoločnosti Kaspersky Labs[11] v roku 2007 ako nahrávku priloženú k emailu. Jednalo sa o variantu Stock spamu, tzn. spamu, ktorý sa snaží svojím obsahom ovplyvniť ceny akcií na burze. Nahrávka obsahuje zmutovaný ženský hlas, ktorý číta propagačný text. Aby bol email čo najmenší tak sú nahrávky silne komprimované, a preto je ich kvalita slabá. Zamestnanci firmy Kasperky Labs, predpovedajú tejto forme spamu krátke trvanie.

Kapitola 3

Boj proti spamu

Bojom proti spamu sa zaoberalo mnoho expertov ako aj spoločností, a tak existuje celá škála produktov, ktoré sa snažia spam čiastočne alebo úplne eliminovať. Bohužiaľ žiadne riešenie neprinieslo úspech, vo forme 100% identifikácie spamu a 0% zle identifikovaného hamu ako spam. Posledných 5 rokov počet spamov rastie lineárne a spolu s tým rastie výška škôd, ktoré tieto spamy spôsobujú.

Hlavným problémom spamu ostáva fakt, že nie je možné určiť, ktorá IP adresa môže pod akou doménou rozosielať emaily. V konečnom dôsledku teda nevieme určiť či daná adresa odosielateľa je skutočná alebo nie.

Nasadenie antispamových techník však môže, zo sebou priniesť aj radu rizík, ako napríklad výpadok poštového servera (dôsledkom preťaženia filtrami obsahov a pod.) alebo terčom DoS útoku.

3.1 Poznaj svojho nepriateľa

3.1.1 Zneužívanie open relay

Open relay je SMTP server nakonfigurovaný tak, že povoľuje externým systémom odosielať poštu na ostatné externé systémy bez obmedzenia. Takéto systémy sú hojne vyhľadávané spammermi a zneužívané na odosielanie spamu. K takémuto hľadaniu sú väčšinou použité napadnuté stroje z botnetu. V 90. rokoch, keď ešte spam nebol takýmto problémom, boli open relay poštové servere vcelku bežné a dokonca aj prednastavené v niektorých poštových serverov. Dnešné MTA nepočítajú s touto konfiguráciou, a teda je prednastavene vypnutá.

Takto nastavené servere umožňujú spammerovi skrývať svoju iden-

titu a ušetrovať zdroje pri posielaní spamu. Avšak vďaka antispamovým technológiám založeným na kontrole hashu prichádzajúceho emailu, musel spammer miesto poslania jedného emailu, ktorý mal najčastejšie v blind carbon copy (skrytý príjemcovia) veľký zoznam príjemcov a tým využíval prenosové pásmo open relay, do každého emailu vložiť nejaký reťazec, ktorý následne zmenil hash. To však znamenalo, že éra zneužívania open relay za účelom ušetrenia zdrojov, prakticky stratila zmysel. Avšak open relay by stále mohli byť zneužité a z toho dôvodu sú uvedené na väčšine blacklistov (DNSBL).

Nakoľko open relay mal svoje opodstatnenie (možnosť odosielať email odkiaľkoľvek), tak sa musel tento prístup nahradiť iným. Medzi tieto "náhrady" patrí zavedenie SMTP AUTH (autentifikácia na úrovni SMTP protokolu), POP before SMTP (ak užívateľ preukáže, že môže sťahovať poštu cez POP tak mu je umožnené posielať poštu cez SMTP), prípadne webové rozhrania pre prístup k pošte a jej odosielaniu.

Základne zásady, ktoré by sme mali dodržať, aby sme náš server nemali nakonfigurovaný ako Open Relay sú

- správy od lokálnych užívateľov sú povolené bez obmedzenia
- správy od autentifikovaných užívateľov sú povolené bez obmedzenia
- správy od nelokálnych užívateľov sú povolené iba do lokálnych schránok

3.1.2 Botnet alebo ideme na to vo veľkom

Botnety sa vďaka svojej rozsiahlosti dajú využiť na nespočet počítačových kriminálnych aktivít, a to za použitia napadnutých počítačov (zombie) tvoriacich tento botnet. Na začiatku sa používali k riadenému Distributed Denial of Service (DDoS) útoku a dnes sa taktiež uplatňujú v roli odosielateľov spamu. Mimo iné môžu tieto zombie prehľadávať web a hľadať emailové adresy, alebo pridávať komentárový spam. Vlastník botnetu dokáže ovládať svoj botnet za pomoci Command and Control centra (C&C), ktoré v minulosti bolo vo väčšine prípadov reprezentované IRC serverom, kde sa jednotlivé boty napojili a prijímali rozkazy. Novodobé botnety sa však uchyľujú k decentralizácii Command and Control centra, pomocou peer-to-peer technológie. Zvyšuje to mieru ukrytia botnetu ako aj počet kontrolovateľných zombie v rámci botnetu. IRC server by vysoký počet (50 000 a viac) nedokázal obslúžiť. Typicky scenár zneužitia botnetu k odosielaniu spamu, kde C&C sa nachádza na IRC serveru, môže vyzerať takto:

- infikovanie počítača
- pripojenie bota na IRC server, kde je pripravený prijímať príkazy
- spammer si objedná prístup do botnetu
- pošle pomocou IRC príkazy, spolu s emailom, ktorý budú všetky boti odosielať
- botnet začne odosielať zadaný spam

Medzi najväčšie botnety patria Storm a Kraken, kde druhý menovaný bol objavený len pred nedávnom[9] a hneď sa stal číslom jedna čo do počtu infikovaných strojov (až 400 000!) a to aj kvôli faktu, že sa mu podarilo infikovať najmenej 50 firiem zo slávneho rebríčka Fortune 500[10]. Botnet je úspešný hlavne vďaka svojím excelentným maskovacím technikám a možnosti nahrávania nových verzií trojana, a tým implementovať na jednotlivé zombie stroje nové záplaty, ktoré spravia sieť ešte neviditeľnejšou.

V roku 2005 sa podarilo dánskym vyšetrovateľom zadržať trojicu ľudí[12], ktorá operovala s botnetom o veľkosti 1 500 000 počítačov. Ten využívali k vykonávaniu DDoS útokov, ale aj ku kradnutiu identít a rozosielanie spamu. K infikovaniu obetí bol použitý Toxbot trojan.

Botnety využívajú komplexné techniky k utajeniu svojej prítomnosti, a to nie len v rámci napadnutého počítača, ale aj navonok. Jednou z takých techník je aj fast flux. Je to DNS technika, ktorá skrýva phishingové stránky, prípadne stránky, na ktorých je uschovaný malware distribuovaný touto sieťou. Technika spočíva v maskovaní kompromitovaného počítača za sériu ustavične sa meniacich proxy serverov. Ďalšou schopnosťou tejto techniky je distribuovaná schopnosť ovládania botnetu (decentralizácia). Vďaka nim je botnet odolnejší voči odhaleniu.

Posledné odhady veľkosti botnetov boli predstavené výskumníkom Joe Stewartom zo spoločnosti SecureWorks[13]. Podľa jeho výskumov prvá 11-ka najväčších botnetov kontroluje spolu vyše milióna zombie strojov, ktoré dokážu rozoslať neuveriteľných 100 miliárd emailov za deň. Botnet Srizbi, ktorý je podľa Stewarta na prvom mieste, sám o sebe dokáže rozoslať 60 miliárd emailov za deň, a to vďaka 315 000 zombie spadajúcim pod tento botnet.

V oblasti boja proti botnetom sa dvom výskumníkom zo spoločnosti TippingPoint podarilo infiltrovať do siete Kraken. Infiltrovali ju do takej miery, že boli schopný odoslať novú verziu trojanu, ktorá by však miesto zlepšenia zabezpečenia trojana, ho odstránila. Avšak tu už ich výskum

skončil. Naskytá sa otázka, či takáto záplata nie je aj napriek svojmu hlavnému cieľu, a to pomôcť, určitým rizikom. Čo ak záplata spôsobí pád počítača alebo, ešte horšie, vymaže užívateľské dáta.

Bohužiaľ niektoré z dnešných botnetov ako Kraken, sú natoľko vyspelé, že ani ochrana vo forme nainštalovanej najnovšej verzia Antiviru, spolu s najnovšou databázou prípadne niektorý z radov Malwareov detektorov nepomáha. Tieto novodobé hrozby, vďaka meneniu binárnych súborov a pokročilej znalosti fungovania antivirov a ostatných detektorov, sa dokážu schovať do takej miery, že ich 80% antivirov nie je schopná detekovať[9]. Ďalším problémom je aj ochrana. Už nestačí dodržiavať pravidlo, ktoré doporučovalo nespúšťať neznáme súbory od neznámych ľudí. Škodlivý software si dokáže nájsť cestu do vášho počítača letmým nahliadnutým stránky a tým zneužiť chybu vo vašom prehliadači. Užívateľ tým pádom nemá ani tušenia, že sa infikoval. Takýto počítač ostáva infikovaný po dlhú dobu.

3.2 Statický Blacklisting

3.2.1 Popis

Na začiatku existoval malý zoznam užívateľov, ktorý odosielali spam. Preto najjednoduchší spôsob v dávnych dobách bol jednoducho vytvoriť blacklist, ktorý obsahoval emailové alebo IP adresy ľudí, ktorý odosielali spam. Tento zoznam si upravoval administrátor sám za pomoci funkcií jeho poštového agenta. Samozrejme tento postup nemal dlhé trvanie, ale v určitej zmenenej forme sa využíva dodnes.

Zaujímavým príkladom je odmietanie emailov, ktoré pochádzajú z inej domény ako .cz. Táto forma blacklistingu využíva fakt, že minimálne percento/promile spamu pochádza z Českej republiky. Samozrejme je táto metóda až príliš drastická, keďže mnoho ľudí ma hostované emaily na zahraničných serveroch (hotmail.com, gmail.com, ..) alebo pracujú pre nadnárodnú spoločnosť (microsoft.com, ge.com, ...). Jedinou možnosťou, ako obísť takýto blacklist, je sa zaregistrovať na niektorom z českých freemailoch a poslať email cez neho. Prípadných zákazníkov samozrejme táto forma môže odradiť a tak je tento typ "ochrany" nemysliteľný pre firemnú sféru, bohužiaľ niektoré firmy ho aj napriek tomu majú.

3.2.2 Úspešnosť

V dnešnej dobe už prakticky nemá zmysel sa pokúšať ručne zablokovať spammerov, neplatí však pre prípady kedy je spammer človek, amatér, ktorý posiela emaily z tej istej emailovej adresy, v tomto prípade je však lepšie to riešiť súdnou cestou (ak je spam v danej krajine postihnuteľný). Avšak existujú pokročilejšie varianty, ktoré spočívajú v dynamických blacklistoch, ktoré sú spravované spoločnosťami zaoberajúcimi sa bojom proti spamu (pre viac informácií viď sekciu 3.3).

V druhom prípade keď sa zablokujú všetky domény, okrem národnej je úspešnosť závislá od typu krajiny. Ak vychádzame zo štatistík obsahujúcich krajiny, z ktorých pochádza spam[20], zisťujeme že je táto metóda úspešná na 99,65%. Avšak ak vychádzame z iných štatistík[21], tak sa úspešnosť tejto metódy zmenší na 92,4%. Hodnoty sú vypočítane na základe počtu spamov pochádzajúcich z Českej republiky. Tieto čísla sú však iba orientačné, keďže neexistujú komplexné celosvetové štatistiky, vďaka ktorým by sa dali tieto hodnoty bližšie aproximovať.

3.2.3 Nedostatky

Hlavným nedostatkom je vysoká miera údržby, ktorá nakoniec je aj tak neúčinná. Spammeri používajú v niektorých prípadoch náhodne vygenerované emailové adresy odosielateľov, takže po jednom použití sa už k nim nevracajú. U IP adries je táto technika o niečo lepšia, ale pri veľkosti dnešných botnetov (obvyklý odosielatelia spamu), je to nadľudská úloha sa o niečo také pokúšať.

V druhom prípade je zrejmé, že hlavným nedostatkom je hlavným nedostatkom radikálne obmedzenie poštových služieb.

3.3 DNS Blocking List

3.3.1 Popis

DNS Blocking List (DNSBL) je technológia založená na overovaní validity IP adries odosielateľov za pomoci technológie DNS. Organizácie bojujúce proti spamu, vytvárajú databázu, IP adries, ktoré sú kvalifikované, ako stroje rozosielajúce spam. Každá takáto organizácia sa riadi určitými politikami. Je na správcovi poštového servera, ktorú si zvolí. Príkladom takýchto databáz sú SpamHaus a DSBL. Tieto sa zároveň momentálne zaraďujú medzi najpoužívanejšie DNSBL služby, a to vďaka svojej spoľahlivosti a miere false-positive.

Princíp celej technológie je jednoduchý, a preto si ho ukážeme na nasledujúcom príklade.

1. Prijatie emailu našim poštovým serverom
2. Vyextrahovanie IP adresy odosielateľa, ktorá sa následne uloží obrátene (192.168.1.1 => 1.1.168.192)
3. Konštrukcia DNS dotazu, ktorý pozostáva z obrátenej IP adresy, ku ktorej je pripojený názov domény, ktorá poskytuje DNSBL služby (1.1.168.192.sbl.spamhaus.org)
4. Ak na dotaz dostaneme odpoveď "Doména neexistuje" (NXDOMAIN) tak to znamená, že adresa nie je uvedená v danom blackliste. Ak však dostaneme potvrdenie o tom, že doména existuje, znamená to, že je zaradená na blacklist.

Technológia sa implementuje do MTA cez špeciálne príkazy v rámci konfigurácie. V postfixu je to napríklad pomocou príkazu *reject_rbl_client*. DNSBL môže byť využitá aj ako nedefinitívna voľba (tj. ak sa IP adresa ocitne na zozname, tak email nie je automaticky odmietnutí), a to implementáciou do analyzátoru obsahu ako napríklad SpamAssassin. Ak sa IP adresa objaví na zozname, váha tohto emailu sa zvýši čím stúpne pravdepodobnosť označenia emailu ako spam.

3.3.2 Úspešnosť

Podľa štatistík[22] je úspešnosť daných služieb DNSBL, závislá od miery agresivity ich politík. V prípade služby XBL (Exploit block List) sa môže jednať o 50% úspešnosť a pri službe PBL (Policy Block List) sa môže jednať o viac ako 60% úspešnosť pri blokacií príchodzieho spamu. SpamHaus je služba zameriavajúca sa na čo najmenší počet zablokovaných legitímnych užívateľov, čo sa im darí vďaka manuálnemu zadávaniu IP adries. Vďaka tomu však aj dosahuje najmenší počet zablokovaného spamu, ale bezpečne odmietne všetky známe zdroje spamu. Táto služba ako jediná by sa dala klasifikovať ako vhodná k použitiu v MTA k zablokovaniu príchodzieho spamu. Pri použití všetkých vyššie uvedených DNSBL služieb je možné odblokovať až 80% spamu.

Výhodou tohto blokovania (ak je definitívne) je fakt, že nám pomáha zredukovať zdroje potrebné na spracovanie spamu. Spamy pomocou DNSBL odmietame v najrannejšej fázy a tým ušetríme čas pri jeho prípadnej analýze filtrami obsahov alebo doručeniu do samotných schránok užívateľov.

3.3.3 Nedostatky

Najväčším problémom je pochopiteľne miera false-positive, ktorá sa odvíja od miery agresivity politiky použitej služby. Niektoré služby nerozlišujú aké kvantum spamu bolo odoslané z danej adresy a tak aj pri náhodnom infikovaný stroja sa môžete dostať na doménu, aj keď ste vírus, ktorý odosielal spam, včasne a rýchlo odstránili. Výber poskytovateľa je však na samotnom administrátorovi. Ak výber podcení a vyberie si príliš "aktívnu" organizáciu, môže v konečnom dôsledku na to doplatiť a to vďaka odmietaniu legitímnych užívateľov. Preto v takýchto prípadoch nie je dobré voliť túto metódu ako definitívnu, ale len ako formu dodatočnej kontroly, pri ohodnocovaní emailu.

Do niektorých databáz sa dostávajú IP adresy, z ktorých nebol zaznamenaný žiaden spam, alebo bol iba v malom množstve. Hlavným dôvodom je, že niektoré DNSBL označujú automaticky IP adresu ako nedôveryhodné, na základe toho ak reverzný DNS záznam ukazuje na dynamickú IP adresu, prípadne ak adresa má špatne nakonfigurovaný poštový server (Open Relay, viď 3.1.1). Tieto praktiky sú napríklad u služieb SORBS DNSBL, SpamCop DNSBL. V druhom prípade sa však jedná o otázku času kedy by daný server bol objavený spammermi a využití k odosielaniu, avšak u dynamických IP adries nastáva problém, keďže niektoré servere majú IP adresu (väčšinou statickú), ktorá spadá pod rozsah dynamickej siete ISP.

Medzi blokované záznamy sa dostávali aj celé rozsahy ISP a to v prípade ak stamaď pochádzalo neúmerne veľké množstvo spamu.

Spammeri aj v tomto prípade dokázali prísť s účinnou technikou, ktorá spočíva v DDoS útoku na DNSBL servere, čo môže mať za následok že všetka pošta je v prípade definitívneho blokovania odmietnutá.

3.4 Hash filtre

3.4.1 Popis

Prvotný pokus o minimalizáciu dôsledkov zle nakonfigurovaných poštových serverov (Open Relay3.1.1) neskôr sa stala z tohto pokusu ďalšia podporná technológia využitá pri analýze emailu. Celý princíp stojí na faktu, že spammer pre odľahčenie svojich zdrojov vytvorí jeden typ emailu, ktorý potom rozosiela cez počítače svojich obetí. Keďže tento email sa môže opakovane objavovať po určitú dobu, vznikol tento systém, ktorý zaznamenáva hashe takýchto emailov do svojej databázy. Hashe poskytujú samotný užívatelia, ktorý obdržali spam. Každý užívateľ má

určitú váhu v závislosti od jeho príspevkov a tak je tento systém čiastočne odolný aj voči prípadným infikovaním databáze spammermi. Ako náhle sa hash spamu objaví v databáze, tak je pre ostatných užívateľov jednoduché označiť prípadný spam na základe vypočítaného hashu. Jedine čo je treba urobiť po príchode emailu a spočítaní hashu je spýtať sa databázy na výskyt daného hashu. Na základe odpovede pridelíme ohodnotenie danému emailu.

Hlavným predstaviteľom antispamovej aplikácie tohto typu je Razor, ktorý bol napísaný Vipulom Ved Prakashom v jazyku Perl. Program existuje v implementácií pre servere, kde sa používa priamo vo filtru obsahu SpamAssassin, ako aj v poštových klientoch. Medzi ďalšie programy tohto typu patrí aj Pyzor, ktorý pôvodne bol iba implementáciou Razora v Pythonu. Neskôr sa vďaka zmenám v protokolu a faktu že Razor server nebol Open Source začal uberať vlastnou cestou a definovaným vlastného protokolu, čím sa nadobro od Razora odčlenil.

Distributed Checksum Clearinghouse (DCC) je ďalšou z implementácii, ktorá je však odolná voči pokuse o zmenu hashu emailu čiastočnou zmenou obsahu správy a to vďaka fuzzy hashom, ktoré sa vyvíjajú spolu so spamom.

3.4.2 Úspešnosť

Spammeri sa na túto technológiu dokázali jednoducho adaptovať avšak aj napriek tomu tieto spamové techniky pre obídenie porovnávania hashu nevyužívajú v niektorých typoch spamov. Príkladom môže byť phishing, kde sa email snaží vyzerať čo najdôveryhodnejšie a tak si nemôže dovoliť generovať obsah.

3.4.3 Nedostatky

Existuje jednoduchá možnosť prekonania tejto "ochrany". K tomuto účelu bol stvorený program Hash buster, ktorý pridáva do emailov náhodne znaky/slová a tým mení ich výsledný hash. Jedine DCC dokáže ako tak odolávať takýmto pokusom o prelomenie tejto techniky.

3.5 Filtrovanie Bayes

3.5.1 Popis

Filtrovanie využíva Bayesovú vetu, ktorá pomáha odfiltrovať spam od hamu na základe analýzy obsahu emailu a z toho vyvodzovať štatistickú pravdepodobnosť či príchodzí email je spam alebo nie. Princíp filtru je založený na rozdielu formy a obsahu spamu a hamu a následnom pravdepodobnostnom vyčíslení. Bayesová veta implementovaná v tomto smere vyzerá takto

$$P(Spam|Sprava) = \frac{P(Sprava|Spam) \ * \ P(Spam)}{P(Sprava)}$$

kde jednotlivé časti formuly, znamenajú

- **P(Spam — Sprava)** pravdepodobnosť, že správa je spam
- **P(Sprava — Spam)** pravdepodobnosť výskytu slov, obsiahnutých v správe, v spamu
- **P(Spam)** pravdepodobnosť, že ľubovoľná správa je spam
- **P(Sprava)** pravdepodobnosť nájdenia slov obsiahnutých v správe v ľubovoľnom emailu

Filter po príchodu emailu využije svoju databázu slov, kde má každé slovo priradenú pravdepodobnosť, k výpočtu pravdepodobnosti.

Po nasadení je potreba náš filter naučiť rozoznávať spam od hamu. K tomu nám slúžia trénovacie programy, ktoré môžu byť v dvoch módoch učenia. Buď budú rozoznávať spam alebo ham. Po zvolený módu predáme programu emaily, ktoré rozanalyzuje (na základe slov) a uloží si ich do databázy. Vzhľadom k povahe filtru je nutné taktiež dodávať korektné emaily (ham), aby sme tým znížili mieru nesprávneho vyhodnocovanie legitímnych emailov.

Bayesovský filter analyzuje napríklad tieto časti emailu

- slová v telách emailov
- hlavičky
- HTML kód (napr. výskyt farieb a ich rozmanitosť)
- frázy

- meta informácie, ako napríklad kde sa daná fráza vyskytla v textu

Bayesovská metóda sa však nedá efektívne uplatňovať v globálnom merítku, vzhľadom k povahe emailovej komunikácie v rôznych spoločnostiach (v lekárskej spoločnosti bude pravdepodobne častejší výskyt slova 'liek' v hamu a tak toto slovo nebude indikovať spam, avšak v bežnej komunikácií sa väčšinou objavuje práve v spamu), je potrebné, aby si každý vycvičoval spamfilter na mieru svojej spoločnosti a poskytoval trénovacím programom zároveň spamy aj hamy. Nemusí ísť však iba o rozdielne zameranie firmy, ale aj o prípadnú jazykovú multikultúru. Vo firmách kde sa nepoužíva angličtina v rámci emailovej komunikácii je väčšina anglicky písaného spamu korektne označená, zatiaľ čo v anglicky hovoriacej firme, môže prechádzať s väčšou pravdepodobnosťou. Opäť ako hlavný faktor tu figuruje miera učenia a veľkosť databáze obsahujúcej váhy jednotlivých slov.

Medzi najpopulárnejšie serverové implementácie Bayesovského filtru patrí SpamAssassin. Mimo Bayesovského filtru využíva taktiež blacklisty, checksumy a ďalšie externé programy k detekcii spamu. Využíva širokú škálu pravidiel, ktoré sa aplikujú na príchodzí email. Väčšina pravidiel má charakter regulárneho výrazu, s ktorým sa porovnáva prijatý email. Výstupom každého pravidla je jeho ohodnotenie emailu. Čím je vyššie, tým je vyššia pravdepodobnosť, že skenovaný email je spam. Nakoniec sa tieto ohodnotenia sčítajú a v závislosti od nastavenia hranice sa označí daný email buď to ako spam alebo sa prepustí bez označenia. Váhy jednotlivých testov sa dajú upravovať a tak si môžeme systém nastaviť na mieru našim požiadavkám. Aj v prípade, že je SpamAssassin nainštalovaný ako globálna systémová služba, umožňuje užívateľom nastavovať špecifické voľby, ako predefinovanie váhy pravidiel, whitelisty a pod.

Existuje možnosť implementácie filtru nielen na poštovom serveru, ale aj priamo v poštovom klientovi (MUA). Možnosť je o to zaujímavejšia, že sme schopný vycvičiť náš filter na mieru našej komunikácii a tak je úspešnosť takéhoto filtru vyššia, než u serverových variant. Medzi programy tohto typu zaraďujeme napríklad Bogofilter, SpamBayes alebo SpamAware.

3.5.2 Úspešnosť

Úspech tejto metódy je priamo úmerný času alebo prostriedkami, ktoré sme venovali vytrénovaniu nášho filtru. V dobre vytrénovaných filtroch dokáže byť táto metóda úspešná.

V prípade automatizácie učenia bayesovského filtru môžeme dosiahnuť bez údržbový systém, ktorý sa bude sám adaptovať na nové techniky využívané k spamovaniu a popritom si zachová nízku mieru false-positive.

3.5.3 Nedostatky

K fungovaniu metódy, je nutná spolupráca ľudí a to za účelom poskytovania korektných údajov pre trénovanie databázy. Pri benevolencii ľudí, ktorý si nechávajú spam v príchodzej pošte, sa môže databáza čiastočne znehodnotiť, týmto spamom.

Druhým spôsobom, ktorý môže spôsobiť znehodnotenie databáze je aj metóda Bayes Poisoning (Otrávenie Bayesovskej Databázy). Spammery dosahujú zníženie efektivity databáze pomocou pridania náhodných alebo dobre volených slov, u ktorých je pravdepodobné, že sa nebudú vyskytovať v spamu. Týmto znehodnotením môžu vznikať postupne štatistické chyby pri vyhodnocovaní emailov. Štatistická chyba prvého druhu vzniká pri označení hamu ako spam a štatistická druhého druhu, pri označení spamu ako ham. Dosiahnutím jednej z týchto chýb zvyšuje spammer svoje šance k úspešnému prelomeniu filtra a tým k zvyšovaniu počtu nekorektne označených spamov ako ham ako aj zvýšenie počtu false-positive (ham označený ako spam).

Adaptácia spammerov dokáže postupne úspešne prekonávať nástrahy bayesovského filtru. Pri každom novom pravidle, alebo po každom naučený filtra technikám spammerov, prídu spammery s niečím novým. Prv to bola zmena farby, ďalej rozmiestnenia textu. Potom nasledovali pridávania k časti validných textov (úryvky z kníh a pod.) medzi reklamný text, neskôr prešli na obrázkový spam (text bol uložený v obrázku a tým pádom neoskenovateľný), a keď začali spam filtre rozoznávať aj ten, začali sa tieto obrázky deformovať, tj. používali rovnakú technológiu ako sa používa v CAPTCHA, ktorá je však určená na obranu proti spammerom.

3.6 Greylisting

3.6.1 Popis

Metóda postavená na princípu zdražovania emailu. Využíva faktu, že spammery nie sú schopný/ochotný posielať jeden email dvakrát, keďže to by činilo zdvojnásobenie ceny za doručenie. Zároveň pri tom kvantu emailov, ktoré rozosielajú sa nezaoberajú návratovými hodnotami na cieľovom poštovom serveru. Celý koncept tejto metódy spočíva v odmiet-

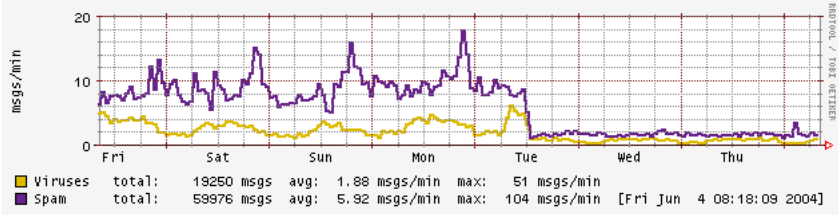
nutí emailu ak kombinácia odosielateľ,prijímateľ a IP adresa odosielateľa ešte nebola zaznamenaná. Email sa však odmieta iba dočasnou SMTP chybou (4xx), takže korektne nastavené poštové servery, dodržiavajúce RFC, sa pokúsia o znovu doručenie po určitom čase. Pri druhom pokuse o doručenie je email povolený a doručený adresátovi.

Medzi najznámejšie greylistingové implementácie zahrňujeme postgrey (implementácia do Postfixu) založená na databáze BerkeleyDB a jej odnož sqlgrey, ktorá vychádza z postgrey avšak operuje nad databázou MySQL.

3.6.2 Úspešnosť

Prínos tohto systému je nepochybný a to nie len kvôli ušetrenia zdrojov nutných na príjem a doručenie spamu, ale aj vďaka nulovej administrácii.

Číselne vyjadriť úspešnosť tejto technológie je však problém a tak nechám za čísla hovoriť obrázok z oficiálnych stránok postgrey 3.1.



Obr. 3.1: Greylisting v praxi, zdroj: postgrey.org

Tejto metóde sa nedá uprieť jedno. Jej jednoduchosť a efektivita. Technológia dokazuje jedno, zdražovanie na spammerov zaberá.

3.6.3 Nedostatky

Oneskorenie medzi prvotným odoslaním a doručením môže trvať aj niekoľko hodín (prednastavená hodnota by mala byť 4 hodiny, avšak väčšina MTA sa pokúsi o doručenie skôr) a tak toto riešenie nie je vhodné do firiem kde je požadované rýchle doručovanie emailov a to za každých okolností. Ako však aj pôvodná špecifikácia emailu upozorňovala, email nie je instantná doručovacia služba. Fakt, že dnes sa však považuje za formu okamžitých správ je spôsobený kvalitou služieb väčšiny poskytovateľov, ktorý sa tak v konkurenčnom boji snažili zvyšovať efektivitu svojich služieb.

Ďalším problémom je však aj samotné odmietanie emailov. Ak email odmietneme tak sa spoliehame na to, že sa poštový server pokúsi o znovu doručenie, nikde si ich dočasne neuchovávame. Avšak nie každý poštový server si dokáže bez chyby poradiť ak obdrží túto dočasnú chybu. V niektorých prípadoch sa tieto emaily nemusia vôbec znovu odosielať (napríklad Microsoft Exchange 2003 SP2[14]).

3.7 DomainKeys Identified Mail

3.7.1 Popis

DomainKeys Identified Mail (DKIM)[16] je metóda emailovej autentifikácie, vychádzajúca z predchádzajúcej špecifikácie, DomainKeys[17]. Pôvodnú technológiu DomainKeys navrhol Mark Delany z firmy Yahoo!. Z tohto návrhu vychádza DKIM. Líšia sa hlavne úrovňou zabezpečenia a vyššou možnosťou konfigurácie[23]. V rámci využívania DKIM na strane odosielateľa je hlavička emailu obalená dodatočnou vrstvou ochrany zahrňujúcou elektronický podpis obsahu emailu, ktorý tam je pridaný odosielajúcim poštovým démonom. Príjemca (MTA) si potom môže overiť podpis prijatej správy. Verejný kľúč odosielateľa je k dispozícii ako súčasť DNS TXT záznamu z odosielateľovej domény. Podpis emailu je tvorený zašifrovaným hashom správy a zakódovaním pomocou base64 algoritmu.

Overovanie emailu prebieha v niekoľkých krokoch

- vytvorenie hashu príchodzieho emailu
- rozšifrovanie podpisu a následne rozkódovanie z bas64
- porovnanie výstupov z predchádzajúcich krokov

Ak sú výstupy rovnaké môžeme považovať takúto správu za legitímnu. V opačnom prípade je pravdepodobné, že sa odosielateľ pokúsil sfalšovať hlavičku, alebo že ma administrátor nekorektne nastavené DNS. Príklad hlavičky emailu, ktorá obsahuje DKIM záznam

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
    d=gmail.com; s=gamma;
    h=domainkey-signature:received:received:message-id:date:from:sender
     :to:subject:mime-version:content-type:content-transfer-encoding
     :content-disposition:x-google-sender-auth;
    bh=ETikCRFd/ht9ee1YuaKbivorc0wwbZ8QdmSv61hjMlc=;
    b=C7gK+AoZqTOwJuDWhqsMTzAVIT4AgubKuC70gBSXmDJm7I0qsNhumx1sz0KjgIQX5+
     lzILJoPBMzuWGLrwqiquI1dZhT3P1vJ56dJlEuPcnMwjc8KQqcEx6yvD1WS3YF4kDyWm
     x7Jq40vD/ufugn/K4RmyEHkd/MRrBDIFsP9wg=
```

Z tejto hlavičky sú najdôležitejšie tieto záznamy

- **a=rsa-sha256** použitý algoritmus

- **d=gmail.com** doména voči, ktorej sa bude overovať
- **s=gamma** selektor domény, ktorý sa používa pre rozlišovanie kľúčov v rámci domény. Výhodou je možnosť definovania rôznych politík pre jednotlivé kľúče definované týmto selektorom.

Je možné popri využívaniu DKIM, používať aj predchádzajúcu verziu, DomainKeys a to v tom istom emailu. Odpovedajúci záznam DomainKeys z rovnakého emailu, z ktorého pochádza DKIM hlavička v predošlom výpise

```
{\tt
DomainKey-Signature: a=rsa-sha1; c=nofws;
    d=gmail.com; s=gamma;
    h=message-id:date:from:sender:to:subject:mime-version:content-type
      :content-transfer-encoding:content-disposition:x-google-sender-auth;
    b=QOUipJdg6m8nPOyA20cYdjHF3HheNLWtIZKTtTUbA8pz4sSkn612GjbPiJuxRlBWe0
      Ks1V6UZQjwu0nMkCHF2z6/OwunmCoaTFYHSJJQt2EV2a4IWVaIqiZfe/KA0W+EpHH3Qr
      zjwtZSZrbwziVkCL2isMDDOvZWmVUk1H5d3Ag=}
```

Pre overovanie sa skonštruuje dotaz na základe vyextrahovanej domény (d=) a selektora (s=). Výsledný dotaz by v našom prípade vyzeral takto

gamma._domainkey.gmail.com

Pri spustení dotazu na takýto záznam dostaneme z DNS odpoveď

```
gamma._domainkey.gmail.com TXT k=rsa; t=y; p=
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDIhyR3oItOy22ZOaBrIVe9m/
iME3RqOJeasANSpg2YTHTYV+
Xtp4xwf5gTjCmHQEMOs0qYu0FYiNQPQogJ2t0Mfx9zNu06rfRBDjiIU9tpx2T+
NGlWZ8qhbiLo5By8apJavLyqTLavyPSrvsx0B3YzC63T4Age2CDqZYA+OwSMWQIDAQAB
```

kde v položke p= je zaznamenaný verejný kľúč v base64 kódovaniu.

3.7.2 Úspešnosť

Zatiaľ jediný úspech, je presvedčenie niektorých veľkých organizácií k podpore tohto mechanizmu. Medzi tieto organizácie patria: Cisco, EBay, PayPal, IBM, Verisign, Yahoo! a ďalšie. Ako súčasť ohodnocovacích algoritmov, dokáže znížiť mieru false-positive ak odosielateľ používa DKIM. Dokáže takisto jednoduchšie detekovať phishingové útoky, prípadne falšovanie hlavičiek emailov.

3.7.3 Nedostatky

Mladá technológia (2007), ktorá nie je moc rozšírená a tak je úspešnosť blokovania spamu na základe podpisu zatiaľ nepoužiteľná. Pri prípadnom rozšírení je pravdepodobné, že sa spammeri pokúsia o útoky na DNS servere, ktoré držia verejné kľúče vo svojich záznamoch.

3.8 Sender Policy Framework

3.8.1 Popis

Sender Policy Framework (SPF), umožňuje takisto ako v prípade DKIM, identifikovať správy pochádzajúcich z adries, ktoré nie sú autorizované k odosielaniu emailov. Využíva k tomu DNS TXT záznamy, ktoré obsahujú informáciu o tom kto je oprávnený posielať emaily z danej domény. Princíp by sa dal prirovnať k DNSBL avšak rozhodovanie je nechané na skutočných DNS serveroch, ktoré má pod kontrolou správca domény. Prihliada sa ku skutočnosti, že iba vlastník domény môže vytvárať a modifikovať DNS záznamy. V rámci záznamu je obsiahnutá informácia o tom, ktoré adresy sú oprávnené zasielať poštu, ale taktiež čo pravidlo, ktoré sa ma uplatniť v prípade

3.8.2 Úspešnosť

Pri masovom rozšírení tejto technológie by sme mohli teoreticky dosahovať vysoké percento blokovaných spamov, blížiace sa k 100. Pritom vychádzame z faktu, že maskovanie IP adries je príliš komplikované a v niektorých prípadoch nereálne (pri overovaní druhej strany).

Takéto nastavenie si však mnoho poskytovateľov nemôže dovoliť, keďže by ich užívateľom nemusela chodiť v niektorých prípadoch pošta. Rieši sa to zjemnením pravidiel a údržbou záznamov v DNS.

Oproti DomainKeys je táto technológia jednoduchšia k implementácií, nakoľko nie je treba riešiť tvorbu a správu kľúčov. Celá inštalácia spočíva vo vytvorení odpovedajúceho záznamu v DNS.

SPF umožňuje ušetriť zdroje a to zamietnutím správy ešte pred jej prijatím.

3.8.3 Nedostatky

Zneužitie DNS serverov spammermi, avšak to im prináša zvýšenú mieru rizika. Bohužiaľ ako aj u DKIM aj u tejto technológie je nutné, aby sa masovo rozšírila a tým znemožnila spammerom posielať nevyžiadanú poštu. Bez spolupráce ostatných správcov domén stráca táto metóda väčšiu časť zo svojej účinnosti. SPF prináša aj určité obmedzenie služieb pre legitímnych užívateľov keďže väčšina providerov ochraňujú svoju sieť pred zablacklistovaným pomocou zablokovania prístup na port 25. Po zavírení počítača sa stáva v niektorých prípadoch súčasťou veľkého botnetu, cez ktorý sa odosielajú spamy. Ak provider zablokuje port 25 znemožní im

aj po infikovaní túto možnosť. Bohužiaľ, zabráni aj odosielaniu pošty na legitímny SMTP server a tak užívateľ musí využiť SMTP server poskytovateľa. Vzhľadom k počtu providerov je nemožné ich pridávať všetkých do whitelistu (nebolo by to ani rozumné, keďže niektorý môžu skutočne byť pôvodcami spamu). Pri odoslaní email cez SMTP server vášho providera vás cieľový poštový server neuvidí v zozname povolených IP adries a tak vás bude klasifikovať ako spammera (záleží od nastavenia domény).

Kapitola 4

Alternatívy

Princíp Challenge/Response systémov nie je ničím novým a preto sa toho už niektorý vývojari chytili a pokúsili sa na jeho princípu postaviť svoj vlastný systém určený pre poštové servery.

4.1 Tagged Message Delivery Agent

TDMA je open source program vytvorený Jasonom Mastelerom v roku 2001. Je napísaný v Pythonu a podporuje tieto MTA: qmail, Postfix, Exim, Courier a Sendmail. Program mimo povoľovania/zakazovania odosielateľov na základe whitelistov a blacklistov implementuje určitú formu Challenge/Response systému. Ten spočíva vo vygenerovaní emailu, ktorý bude vyzývať užívateľa, aby poslal ľubovoľný email na špeciálne vygenerovanú verifikačnú adresu. Tá je taktiež nastavená v hlavičke "Reply-to", takže užívateľ v konečnom dôsledku potrebuje iba využiť funkciu "Odpovedať" vo svojom poštovom klientovi a nemusí žiadnym spôsobom meniť obsah emailu, ale môže ho rovno zaslať.

Špeciálna emailová adresa je vygenerovaná z pôvodnej užívateľskej adresy, ku ktorej sa pridáva špeciálna prípona. Adresa môže vďaka tejto prípony nadobúdať rôzne významy.

- Adresa s časom
  Označenie adries časovým razítkom, ktorý obmedzuje platnosť tejto adresy do určitého dáta, ktoré je zakódované v prípone. Tieto adresy sú vďaka tomu vhodné k jednorázovým účelom, keď požadujeme interakciu na fórach alebo diskusných skupinách, ktoré však odhaľujú našu adresu.

- Adresa s odosielateľom Emailová adresa obsahuje vo svojej prípone

zakódovaného odosielateľa správ, ktorý ako jediný bude môcť zasielať emaily na túto adresu. Ostatný odosielatelia, ktorý sa budú pokúšať poslať email na túto adresu budú presmerovaný na klasický verifikačný proces TMDA.

- Adresa s kľúčovým slovom Emailová adresa obsahuje vo svojej prípone zakódované kľúčové slovo. Pri príchode emailu na túto adresu sa bude kontrolovať prítomnosť kľúčového slova v adresa odosielateľa. Keď sa nenájde, email sa dostane do klasického verifikačného procesu.

Systém prináša skvelé výhody v oblasti jednoduchosti a prívetivosti smerom k užívateľovi. Užívateľ nemusí riešiť žiadnu matematickú úlohu ani odpovedať na otázky, stačí mu len odpovedať.

V prípade špeciálne vygenerovaných adries, systém prináša rozšírenie využiteľné v mnohých aspektoch emailovej komunikácii. Napríklad pri komunikácií s internetovým obchodom si vytvoríme adresu s kľúčovým slovom obsahujúcim doménu internetového obchodu a s touto adresou sa potom zaregistrujeme cez webový formulár. Mechanizmus nás ochráni v prípade nesolventných obchodov, ktoré predajú našu emailovú adresu tretej strane, ktorá ju zneužije na spamovanie. Ten istý mechanizmus však môžeme využiť pri registrácií v diskusních skupinách.

Vzhľadom k dlhému vývoji, ponúka toto riešenie kompletnú škálu riešení, ktoré chránia užívateľov pred spamom avšak tieto riešenia sa dajú obísť ak sa spammerom podarí naimplementovať dostatočne inteligentní mechanizmus, ktorý bude odpovedať na vygenerovanú emailovú adresu. Problémom sú však aj samotné poštové servery, ktoré môžu automaticky povoliť spammera.

To môže nastať ak spammer odošle email pod falošnou adresou odosielateľa, ktorá však bude smerovať na živý poštový server a my sa na ňu pokúsime odpovedať našim verifikačným emailom. Týmto odoslaním na falošnú adresu však dorazí na živý poštový server, ktorý môže za určitých okolností vyhodnotiť náš email ako nedoručiteľný (neexistujúca adresa a pod.) a vrátiť nám ho späť. Týmto návratovým chybovým emailom však splnil požiadavky dané systémom TMDA. Keďže si autor bol vedomý tohto rizika tak umožnil meniť vlastnosti týchto verifikačných emailov, aby sa takéto prípady mohli eliminovať alebo aspoň zredukovať.

TMDA končí pri analýze adries, zatiaľ čo program Spampuzzle, ktorý je výsledkom tejto práce, tam iba začína. Spampuzzle je založený na analýze samotného obsahu email. Podľa konfigurácie sa dá nastaviť miera analýzy emailu a tým aj miera zaťaženia poštového servera.

Spampuzzle naproti TDMA vyžaduje väčšiu interakciu od užívateľa. Výhodou väčšej interakcie je však menší počet spamu. V prípade automatického spamu sa číslo blíži k nule.

Kapitola 5

Spampuzzle

Cieľom diplomovej práce nebolo len zmapovať situáciu na trhu antispamových riešení, ale taktiež pokúsiť sa na základe nadobudnutých znalostí skonštruovať antispamové riešenie, ktoré sa pokúsi využiť faktor zdražovania emailu ako hlavným pilier.

Greylisting je v dnešnej dobe jednou z najúspešnejších riešení, ktoré stojí na faktu zdvojnásobenia počtu vynaložených zdrojov k odoslaniu emailu. Pri paranoidnejších konfigurácií môže byť využitie zdrojov ešte niekoľko krát väčšie, nakoľko spammer nevie aká je nastavená doba medzi prvým zdržaním emailom a prvým prepusteným emailom. Avšak postupom času sa cena za odoslanie jedného emailu zmenšuje a to vďaka zvyšovaniu kapacity internetových liniek a zvyšovaniu počtu užívateľov, ktorý môžu byť využití ako zombie v rámci botnetu. Preto je treba myslieť dopredu a pokúsiť sa navrhnúť riešenie, ktoré už teraz bude požadovať vysokú cenu za doručenie od odosielateľa a tým odradí prípadných spammerov, avšak nie tak vysokú, aby odradila aj prípadných legitimných odosielateľov.

Výsledok tejto práce sa pokúša eliminovať automatizovaných botov odosielajúcich spamy ako aj nežiadúce osoby a to využitím konceptu Client Puzzle, ktorý však nebolo možné implementovať transparentne do takej miery, aby neboli užívatelia obťažovaný. Systém postavený na čiastočnej modifikácií tohto Client Puzzles protokolu sa nazýva ChallengeResponse systém.

Projekt Spampuzzle sa snaží o aktívne blokovanie nevyžiadanej pošty. Hlavným cieľom bolo skonštruovať systém, ktorý bude plne kompatibilný so súčasnými poštovými serverami, so zameraním na konkrétnu implementáciu poštového serveru, Postfixu. Zároveň k jeho funkčnosti nebudú vyžadované žiadne ďalšie rozsiahle nástroje ako webový server, SQL databáza a pod. Riešenie bolo navrhnuté s čo najväčším ohľadom

na jednoduchosť implementácie, aby mohlo byť jednoducho nasadené aj do ostatných poštových serverov.

5.1 Použité technológie

5.1.1 BerkeleyDB

BerkeleyDB je open source, embedded databázová knižnica. Výhody spojené s využívaním tejto knižnice, je hneď niekoľko

- stabilita
- rýchlosť
- popularita
- plná podpora ACID
- vysoká miera škálovateľnosti vďaka implementácii sofistikovaného replikačného mechanizmu
- nulová administrácia

Výhody sú však sprevádzané určitými nevýhodami

- nie je možné sa cez sieť vzdialene pripájať na databázu
- nepodporuje SQL jazyk
- prístupy k databázy nepodliehajú žiadnemu užívateľskému konceptu

Ani jeden z nedostatkov nás však nebude pri práci s BerkeleyDB v rámci Spampuzzle obmedzovať. Spampuzzle obsahuje databázu na serveru kde beží a pristupuje k nemu iba on, čiže nie je potrebná implementácia komplexného autorizačného konceptu. Stačí využívať autorizačný koncept filesystému, aby obsah databáze ostal utajený.

Jazyk SQL by bol v konečnom dôsledku pre nás jedine príťažou, keďže nepotrebujeme konštruovať žiadne komplexnejšie dotazy. Embedded databáze umožňujú vytvárať vlastné rozhranie pre prístupu k dátam. Implementácia vlastného rozhrania nám umožňuje zvýšiť efektivitu pri práci s databázou.

Základom dátovej reprezentácie v embedded databáze je kombinácia kľúč:hodnota. Databáza neobsahuje definíciu schématu tabuliek ani

žiadne stĺpce. Je možné vyhľadávať záznamy iba za pomoci kľúča, tzn. že nie je možné hľadať podľa hodnoty poľa 'hodnota'. V prípade, že daný kľúč existuje v tabuľke, tak je k nemu priradené pole hodnota. Pole nemá definovaný formát a tak pri viac atribútových hodnotách, si musíme zvoliť vlastný formát zápisu, ktorý bude zohľadňovať tento fakt (popis Spampuzzle formátu je v časti 6.2).

5.1.2 Client Puzzle Protokol

Koncept protokolu client puzzle bol predstavený firmou RSA Security Inc.[15], ktorá sa v roku 2000 zaoberala jeho využitím na poli DNS a Web serverov.

Protokol vyžaduje po každom z klientov, aby vyriešil matematickú úlohu. V opačnom prípade mu nebude umožnené využívať službu na serveru. Po jeho vyriešení je klient oprávnený využívať požadovanú službu.

V prípade legitimnej relácie je táto operácia pre klienta bez problémová (v zásade spotrebuje minimálne percento zdrojov, v tomto prípade CPU). Avšak v prípade užívateľa, ktorý sa pokúša masívnym zasielaním požiadaviek na server obmedziť jeho prevádzku, je táto úloha náročná na jeho zdroje a tým mu znemožní obmedziť našu prevádzku. Nebezpečný užívateľ, bude musieť vynaložiť veľké množstvo zdrojov na to, aby sa mu podarilo každú matematickú úlohu vyriešiť.

Na prvý pohľad je elektronická pošta vhodná na takúto implementáciu ale bohužiaľ, je veľa faktorov, ktoré znemožňujú transparentnú implementáciu tohto protokolu

- príliš jednoduchý návrh SMTP protokolu neumožňuje v dostatočnej miere rozširovanie o ďalšie funkcionality
- existencia viacerých implementácií poštových serverov, ktoré ani v týchto dobách nepodporujú protokol v plnej miere
- nutnosť aktualizovať všetky inštalácie poštových serverov
- široká škála poštových klientov, ktoré by sa museli prispôsobiť prípadnej zmene protokolu

Implementáciu protokolu bolo nutné spraviť bez zmeny SMTP protokolu ako takého, ako aj bez zmeny jednotlivých poštových agentov. Riešenie teda muselo vyžadovať interakciu od užívateľa. Komunikácia medzi nimi však naďalej prebieha výhrade cez elektronickú poštu.

5.1.3 Simple Mail Transfer Protocol

Simple Mail Transfer Protocol (SMTP)[18], je nosným protokolom elektronickej pošty. Využívajú ho poštové klienti k odosielaniu a poštové servery k prijímaniu a odosielaniu emailov. Pre prijímanie pošty bežným užívateľom slúžia protokoly ako POP a IMAP.

SMTP je jednoduchý textový protokol, ktorý operuje nad obmedzenou sadou riadiacich príkazov. Základná sada príkazov s popisom je v tabuľke 5.1.

Príkaz	**Argument**	**Popis**
HELO	*<FQDN>*	Identifikácia klienta voči serveru. Ako argument sa udáva fully-qualified domain name (FQDN) klienta.
EHLO	*<FQDN>*	Identifikácia klienta voči serveru s podporou rozšírených funkcií SMTP protokolu. Ako argument sa udáva fully-qualified domain name (FQDN) klienta.
MAIL FROM:	*ex1@example.net*	Definovanie adresy odosielateľa, ktorá je zadaná ako argument tohto príkazu.
RCPT TO:	*ex2@example.net*	Definovanie adresy prijímateľa, ktorá je zadaná ako argument tohto príkazu.
DATA	*<EMAIL>*	Označenie konca "obálky" a začiatku samotnej správy. Táto sekcia sa ukončuje zadaním '.' na prázdny riadok.
RSET	N/A	Resetuje reláciu a vymaže všetky zásobníky, ktoré udržiavajú informácie o vykonaných nastaveniach a operáciách (MAIL FROM, RCPT TO, DATA)
HELP	N/A	Zobrazí pomoc
QUIT	N/A	Ukončenie relácie

Tabuľka 5.1: Základná sada SMTP príkazov

Časti *MAIL FROM* a *RCPT TO* tvoria obálku (envelope) a časť *DATA* obsahuje samotnú správu. Správa musí obsahovať hlavičku, ktorá

obsahuje dodatočné informácie ako napríklad o emailovej adrese príjemcu, kódovaní, výstupu zo spamového filtra a ďalšie. Informácie v hlavičke sú zapísané vo formáte

Typ: Informácia

Medzi najznámejšie typy informácii v hlavičke patria

- From
  Adresa odosielateľa alebo jeho meno, ktoré uvidí prijímateľ v kolónke "Od:". V prípade, že nie je zadefinovaná, tak sa použije adresa z obálky (MAIL FROM). Pole môže obsahovať ľubovoľný text, avšak potom nie je legálne podľa RFC 2822 a preto môže byť odmietnutá niektorými poštovými serverami.

- To
  Adresa prijímateľa alebo jeho meno, ktoré uvidí prijímateľ v kolónke "Komu:". Ak táto položka nie je zadefinovaná, tak sa u príjemcu zobrazí v kolónke "Komu:" text undisclosed-recipients, čo v zásade znamená skrytého príjemcu. Pole môže obsahovať ľubovoľný text, avšak potom nie je legálne podľa RFC 2822 a preto môže byť odmietnutá niektorými poštovými serverami.

- Reply-To
  Adresa, na ktorú má byť smerovaná odpoveď. Nemusí byť zhodná s pôvodnou adresou odosielateľa

- Cc
  Carbon copy, obsahuje zoznam emailových adries, ktorým bude doručená kópia správy, aj keď správy nemusí týkať ich osoby. Dá sa to prirovnať k diskusii medzi dvoma ľuďmi, kde sú zapojený aj prísediaci. Tí figurujú, práve v Cc zozname.

- Bcc
  Blind Carbon Copy, slepá kópia, kedy adresát vidí iba adresy uvedene v poli To: prípadne Cc: ale nie zoznam adries uvedený v Bcc. Pritom na adresy v uvedené v tomto zoznamu je doručená kópia tejto správy.

- Subject
  Predmet správy.

- Date
  Dátum kedy bola správa napísaná.

- Message-ID
  Identifikácia emailu v rámci poštového démona. Toto identifikačné číslo by malo byť v rámci celého Internetu jedinečné, ale sú prípady keď nie je.

- References
  Message-ID správ, ktoré boli poslané v rámci tejto konverzácie. Prvé ID je najstaršie (prvý email v konverzácii) zatiaľ čo posledné ID je predchodca tohto emailu

- `Prázdny riadok`
  Označenie konca "obálky" a začiatku samotnej správy. Táto sekcia sa ukončuje zadaním '.' na prázdny riadok.

Obr. 5.1: SMTP konverzácia

```
220 mail.localdomain ESMTP Postfix
HELO client1.example.net
250 mail.localdomain
MAIL FROM: <ex1@example.net>
250 2.1.0 Ok
RCPT TO: <ex2@example.net>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: EX1 <ex1@example.net>
To: EX2 <ex2@example.net>
Subject: Testing email
Reply-to: <ex1@example.net>

Good morning,

        this is just test email.

                EX1
.
250 2.0.0 Ok: queued as A7F62189C81
QUIT
221 Bye
```

Protokol pri zadaní príkazov odpovedá kódmi, ktoré symbolizujú status vykonaného príkazu. Návratový kód je symbolizovaný troma ciframi, kde prvá udáva celkový status. Ďalšie cifry sú použité k upresneniu statusu, ktorý nastal.

Zoznam používaných označení pre celkový status:

- 2xx = Úspešné vykonanie príkazu
- 3xx = Server čaká na ďalšie dáta
- 4xx = Dočasná chyba

- 5xx = Trvalá chyba

Klient inicializuje spojenie na server (štandardne server počúva na porte 25) a posiela počas relácie kontrolné príkazy. Pri inicializovaní spojenia s poštovým serverom sa klient predstaví príkazom HELO prípadne ak chce mať možnosť využívať rozšírenú sadu príkazov, tak použije EHLO. Následne zadá odosielateľa resp. prijímateľa a to cez príkazy MAIL FROM resp. RCPT TO. Príkazom DATA oznámime, že ďalšie riadky budú obsahovať samotný email.

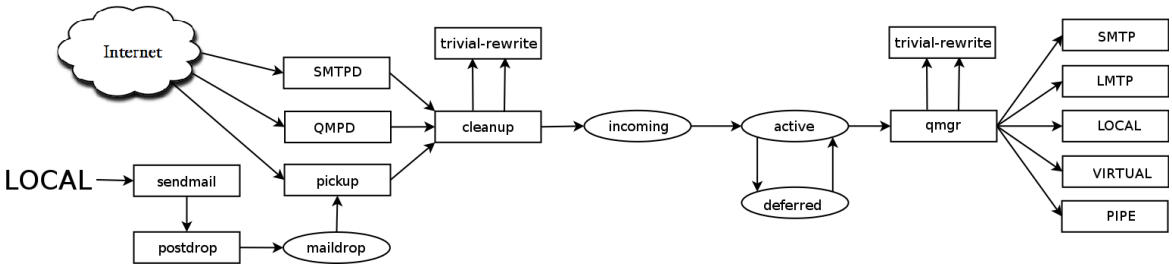
Ukážkové odoslanie emailu cez SMTP protokol je uvedené v príklade 5.1. Riadok začínajúci trojciferným číslom reprezentuje výstup poštového servera. Všetky ostatné riadky reprezentujú klientský vstup.

5.1.4 Postfix

Postfix[19] je open source program určený na prenos elektronickej pošty z jedného počítača na druhý. Programy tohto typu sú označované ako Mail Transfer Agent (MTA). Postfix vyvinul bezpečnostný expert Wietse Venema, ktorý zúžitkoval svoje skúsenosti a použil ich pri návrhu. V dobe keď sa Wietse rozhodol vyvinúť tento systém neexistovali podľa neho žiadne bezpečné a dostatočne flexibilné MTA na trhu. O tom svedčili hlavne bezpečnostné nedostatky vtedy veľmi využívaného MTA Sendmail. Väčšina týchto nedostatkov bola kritických a tak umožňovala prebrať kontrolu nad systémom správne naformulovaným vstupom, vďaka pretečeniu zásobníku. Medzi takéto chyby patrili [3] alebo [4].

Postfix je navrhnutý s čo najväčším ohľadom na bezpečnosť, jednoduchosť, rýchlosť a spoľahlivosť. Hlavným špecifikom Postfixu je hlavne jeho modulárny návrh, vďaka ktorému bol Wietse schopný v dostatočnej miere splniť svoje požiadavky. Každý modul beží s minimálnymi právami, ktoré stačia k vykonaniu úloh, ku ktorým bol určený. Jednotlivé moduly sa dajú podľa potreby zapínať a vypínať a tým sa chrániť pred zneužitím cez službu, ktorú ani nevyužívame. Moduly sú v prednastavenom prostredí nastavené ako chrootované. Vďaka chroot môžeme vytvoriť uzatvorené prostredie(v rámci špecifikovaného podadresára), v ktorom beží program. V prípade ak by sa našla bezpečnostná chyba v jednom z modulov, tak bezpečnosť ostatných nie je rovno kompromitovaná. Jednotlivé moduly a ich prepojenie v rámci architektúry postfixu môžeme vidieť na obrázku 5.2, kde v obdĺžniku sú programy postfixu a v ovále sú postfixové fronty.

Podľa štúdie firmy SecuritySpace[1] používa až 85% poštových serverov niektorý z týchto MTA: Postfix, Exim, Sendmail, Microsoft Exchange.



Obr. 5.2: Architektúra Postfixu

Podľa ďalšej štúdie[2] je tento trh viac diverzifikovaný, ale aj napriek tomu tam figurujú všetky hore spomenuté MTA.

Spampuzzle je implementovaný do postfixu ako after-queue content filter. To znamená, že analýza obsahu sa vykonáva až po prijatí emailu. Tým však strácame možnosť zahodiť email pred prijatím a tým ušetriť zdroje poštového serveru. Neostáva nám nič iné ako si email v prípade potreby ukladať u seba na disk (pri nastavenej operácii HOLD). Zvolením druhej varianty, a to before-queue content filter, by sme naopak mohli pri obrovskom zaťažený alebo veľkých emailoch nebyť schopný prijímať emaily následkom spomalenia spôsobeného analýzou emailu Spampuzzlom.

Avšak aj napriek tomu je v prípade potreby možné nasadiť Spampuzzle ako before-queue content filter a to za pomoci SMTP proxy, ktorá bude preposielať prijatý email do nášho démona.

5.2 Problémy a doporučenia

Otázky tvoria základ celého konceptu Spampuzzle. Na nich to stojí ale zároveň aj padá. Je dôležité voliť také otázky aby ich mohli ľudia, s ktorými chceme komunikovať zodpovedať. Avšak v prípade, že zvolíte otázku, na ktorú je príliš ľahká odpoveď, ochrana nemusí zafungovať, čo bude mať za následok prepúšťanie spamu. Jednoduchosť z pohľadu odpovede je v predchádzajúcej vete úmyselná. Akokoľvek ťažkú otázku vymyslíte, je dôležité aká je odpoveď. Ak je odpoveď napríklad 2008 (čo predstavuje tento rok), môžete pri určitom nastavení Spampuzzle (napr. kontrola bude vykonaná naprieč celým emailom), môže sa stať, že aj robot na ňu dokáže odpovedať. Predstavme si situáciu kedy odpoveď na otázku je spomínané číslo 2008. Príde nám spam (od spammera so skutočnou adresou alebo od užívateľa, s ktorým nechcem komunikovať), na

ktorý odpovedáme štandardne emailom s otázkou, ak nie je povolený. Užívateľ alebo spammer, sa pokúsi odpovedať na našu otázku a to tak, že nám iba odpovie na náš email a súčasťou odpovede bude aj citácia (ako to robí väčšina MUA) alebo hlavička predchodzieho emailu, v ktorej je obsiahnutý aj dátum. Tu však nastáva problém, keďže v dátum sa bude zhodovať s odpoveďou a spampuzzle to bude považovať za korektnú odpoveď a tým povolí užívateľa. Spammeri však väčšinou využívajú špeciálne programy na odosielanie emailov, ktoré rozhodne ani nasadením takýchto systémov nezačnú posielať takýto reply email, ktorý bude citovať predchádzajúci email. Preto je dôležité mať toto pravidlo hlavne na vedomí ak nechcem aby nám mohol napísať niekto kto využíva klasické prostriedky emailovej komunikácie.

Spampuzzle vďaka spôsobu fungovania dokáže mimo spam limitovať aj poštu od ľudí, s ktorými sa nechceme baviť. Príkladom by mohol byť profesor na fakulte MFF UK, ktorý chce prijímať emaily iba od študentov MFF UK. Zvolí si otázku, na ktorú vie odpovedať každý študent, ale už nie každý neštudent(najideálnejšie by bolo ak by to vedeli IBA študenti, ale takéto verejné tajomstvo je ťažko udržateľné). Napríklad aké by mal profesor prihlasovacie meno v prípade že by bol prijatý na univerzitu v roku 2003 a nenastala by žiadna kolízia s iným menom. Na základe mena profesora(ktoré vieme) a dodatočného špecifikovania podmienok vieme vytvoriť prihlasovací login. Keďže každý študent vie ako sa vytvára takýto login, nie je preňho problém túto otázku zodpovedať. Avšak v prípade ak sa mu pokúša napísať niekto mimo univerzitu neuspeje, alebo bude musieť poznať kolorit univerzity (čiže bude mať niečo s ňou spoločné), alebo bude musieť vynaložiť enormne úsilie k tomu aby niečo také zistil.

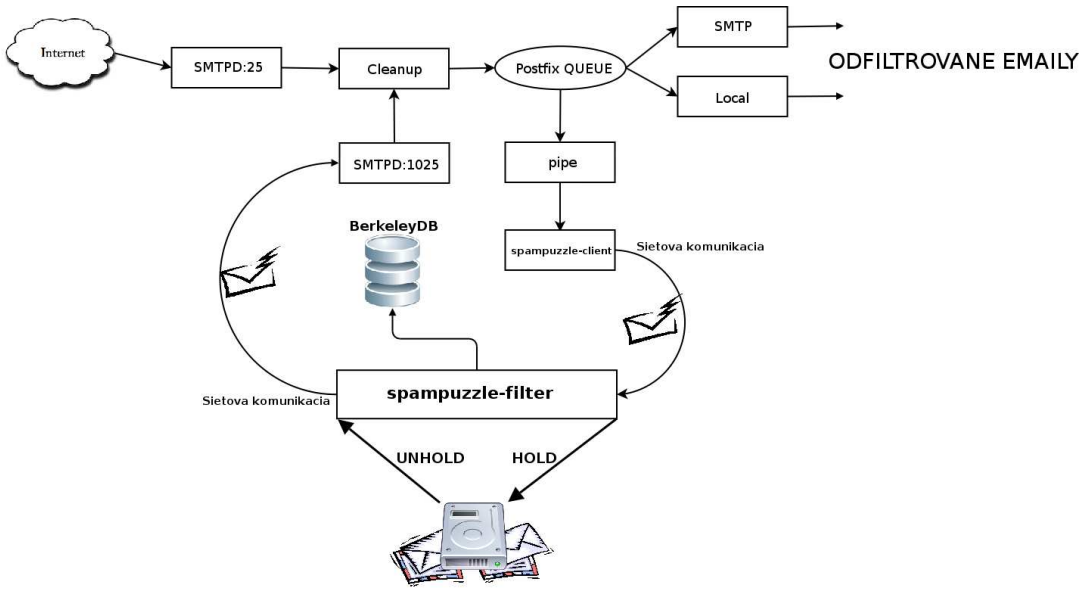
Ďalšia situácia, ktorá môže nastať je, že spammer odoslal email z falošnej adresy. My sa na ňu pokúsime odpovedať otázkou avšak email buď to nedorazí vôbec alebo cieľový poštový server nebude poznať užívateľa, ktorému to má doručiť. V tomto prípade sa email vráti späť odosielateľovi, tj. nám, ako bounce email. V ňom budú obsiahnuté dôvody nedoručenia. Tento email dorazí pri štandardnej konfigurácií z adresy mailer-daemon@<domena.cz>. Preto je dôležité mať práve túto adresu vo whitelist databáze aby ste sa vyvarovali blokacií týchto emailov, keďže sa môžu vyskytovať aj pri legitímnej konfigurácií.

Taktiež je rozumné nastavovať rozumný interval medzi odosielaním opakovaných otázok, v prípade ak sa užívateľovi nedarí odpovedať správne. Horný limit počtu opakovaných zaslaní otázky by taktiež nemal byť vysoký. Oneskorenie pri opakovaných otázkach je nastavené defaultne na

3600 sekúnd a maximálny počet odoslaných emailov je 3.

Diskusné skupiny sú pre Challenge/Response systémy vždy veľký oriešok. Keďže v závislosti od nastavenia diskusnej skupiny môžeme buď to dostávať email akoby priamo od diskusnej skupiny alebo od samotného tvorcu príspevku. Prvý spôsob je kompatibilný s naším systémom a tak ho nie je treba nijak špeciálne ošetrovať. Čo je jedine potrebné je zadať diskusnú skupinu do whitelist databáze. V druhom prípade však nastáva problém, keďže je príliš náročné sledovať fluktuáciu ľudí v danej skupine a pridávať/odoberať ich. Spampuzzle rieši tento problém pomocou analýzy hlavičky a hľadaniu časti *List-Post*, ktorá by mala definovať z ktorej diskusnej skupiny email prišiel. Problém je jedine v slovnom spojení "mala by". Niektoré diskusné systémy túto hlavičku nevypĺňajú a tak vám neostáva nič iné ako povoľovať každého užívateľa zvlášť. Ak však táto hlavička je vyplnená Spampuzzle si vezme jej hodnotu (mala by obsahovať emailovú adresu diskusnej skupiny) a použije ju miesto pôvodného odosielateľa pri kontrole.

5.3 Architektúra



Obr. 5.3: Architektúra spampuzzle a integrácia do Postfixu

Základnú kostru projektu Spampuzzle, tvoria dva moduly, z ktorých je jeden klient (nazývaný taktiež Slave) a druhý server (nazývaný taktiež

Master). Slave je reprezentovaný jednoduchým shell skriptom, ktorý je spjatý s Postfixom. Postfix predáva prijaté emaily (spolu s hlavičkami) práve tomuto skriptu za pomoci postfixového príkazu pipe. Ako dáta v tejto transakcii figurujú emaily.

Slave k emailu pridelí vlastnú hlavičku. Tá slúži k predávaniu informácií Mastrovi. V samotnej hlavičke pôvodného emailu nemusia byť niektoré dôležité informácie obsiahnuté (informácia o SASL), prípadne môžu byť pozmenené (From:, To:).

Slave následne pošle nepozmenený email spolu s našou hlavičkou, cez sieťové rozhranie. Po zaslaní emailu skončí spracovanie Slavea. O doručenie emailu sa teraz postará Master.

Druhý modul je reprezentovaný Mastrom, ktorý plní rolu démona prijímajúceho emaily na preddefinovanom porte (určuje sa pomocou argumentov pri spúšťaní Mastera) a spracováva ich. V rámci Spampuzzle Master figuruje ako serverová časť. Popis nastavení Mastera nájdete v sekcii 5.5.

Master považuje všetok vstup na svojom porte v rámci jednej relácie za jeden email. Akonáhle druhá strana dokončí transfer Master začne spracovávať email a na základe získaných informácií z hlavičky, ktorú mu poskytol Slave sa rozhodne ako naloží s daným emailom. Môže ho buďto odložiť na disk, pozmeniť predmet a poslať, prípadne rovno poslať bez akejkoľvek zmeny.

Aby sa nám nevytvoril cyklus v našom poštovom serveru, tak musíme emaily odoslané z Mastera posielať cez druhého SMTP démona (defaultne počúva na porte 1025 a definujeme si ho v rámci inštalácie, viď 5.4). Tento druhý SMTP démon však nesmie obsahovať ako filter obsahu náš spampuzzle keďže tým by sa nám vytvoril cyklus. Vďaka tomuto mechanizmu viacerých SMTP démonov s rôznymi nastaveniami môžeme spampuzzle integrovať do už existujúcich poštových systémov, ktoré využívajú aj iné spôsoby filtrácie pošty a tým vytvoriť účinnú reťaz filtrov obsahu.

Výhodou architektúry klient/server je fakt, že môžeme klienta nahradiť niektorým z poštových serverov, ktoré umožňujú posielanie obsahu do externých programov, prípadne môžeme medzi poštový server a Spampuzzle umiestniť SMTP proxy, nakoľko špecifikum poštového serveru je preposielanie emailu na ďalší poštový server (relay). Týmto mechanizmom sa dá Spampuzzle naimplementovať aj do poštových systémov ako Microsoft Exchange a ďalších.

Program interne pracuje v unicode režimu, vďaka ktorému je schopný spracovávať email v rôznych jazykoch. Samozrejmosťou je podpora iných

kódovaní, ktoré sa môžu vyskytnúť v emailoch. Výstupné emaily Spampuzzle (otázky, informácie) sú v unicode avšak kódovanie príchodzích emailov ostáva nezmenené.

Architektúra je načrtnutá na obrázku 5.3, kde je vidieť ako sa Spampuzzle integruje do Postfixu.

5.3.1 Slave modul

Modul slúži ako prostredník medzi postfixom a Mastrom. Jeho hlavnou úlohou je transport emailov z postfixu do Mastra. Mail príjme od postfixu cez jeho štandardné rozhranie pipe (rúra). Rozhranie je vlastnosťami totožné s rozhraním, ktoré poskytuje shellovská variantu rúry ('|'). Štandardný výstup postfixu (STDOUT) je poslaný na druhu stranu rúry kde je náš klientský modul Slave, ktorý dostane email na svoj štandardný vstup (STDIN). Slave nie je perzistentní, ale je spúšťaný pri každom prijatí emailu samotným postfixom. Táto vlastnosť je samozrejme v prípade väčších programov nežiadúca, keďže inicializačná fáza môže trvať viac ako analýza samotného emailu. To bol aj dôvod prečo sú v rámci Spampuzzle 2 moduly.

Inicializácia Modul nemá žiadnu inicializačnú fázu.

Výstup Údaje o odosielateľovi, prijímateľovi, IP adrese odosielateľa ako aj užívateľskom mene SASL získava Slave od Postfixu za pomoci Postfixového programu pipe. Ten dokáže tieto hodnoty o Postfixu získať a predať nám ich ako vstupné argumenty pri štarte nášho skriptu. Postfix dodáva hodnoty, ktoré získal pri SMTP relácií.

Hlavička, ktorá sa pripája na začiatok správ a predáva sa Mastrovi je vo formátu

X-Legit: *odosielateľ*:*prijímateľ*:*IP adresa odosielateľa*:*SASL užívateľské meno*

Za týmto riadkom nasleduje jeden prázdny riadok a to z dôvodu oddelenia našej hlavičky od štandardných hlavičiek emailu.

V prípade, ak užívateľ nie je autentifikovaný cez SASL (takže nemá žiadne užívateľské meno v rámci SASL) je položka "SASL užívateľské meno" nevyplnená. V opačnom prípade sa daná správa pokladá za privilegovanú a podlieha špeciálnym kritériám ako, automatické povolenie komunikácie medzi odosielateľom a prijímateľom a taktiež v prípade ak

sa v predmete objaví špeciálny argument, je tento email považovaný za konfiguračný.

K tomu aby bol mechanizmus privilégií aktivovaný, však nie je treba SASL. Stačí mať definovaný rozsah adries vo whitelist-ip, pre ktoré chcete povoliť konfiguráciu.

Po vytvorení hlavičky a obdržaní emailu zo štandardného vstupu sú tieto údaje za pomoci programu netcat odoslané nášmu Masterovi. Týmto sa úloha Slaveu v procesu doručovania končí. Všetko ostatné už zaobstaráva serverový modul Master.

Ukážkový výstup z tohto modulu, ktorý sa vo výsledku posiela Masterovi, môže vyzerať takto

```
X-Legit: alice@example.net:bob@example.net:192.168.1.1:

From: Alice <alice@example.net>
To: Bob <bob@example.net>
Subject: Hello
Content-Type: text/plain; charset=UTF-8
MIME-Version: 1.0
Content-Transfer-Encoding: 8bit
Message-Id: <20080808053954.0D41C189DAA@mail.localdomain>
Date: Fri,  3 Aug 2008 17:39:54 +0200 (CEST)

Hi Bob

Alice
```

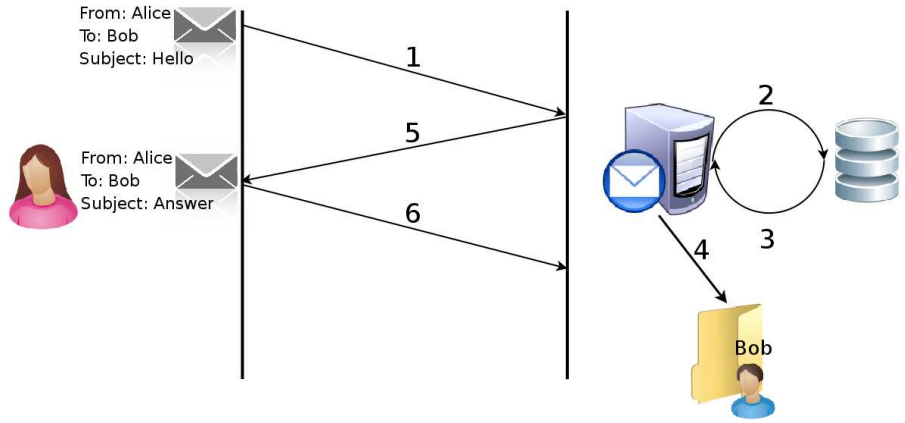
5.3.2 Master modul

Základnou kostrou celého programu tvorí tento modul napísaný v Perlu, ktorý príjme emaily cez sieťové rozhranie a na základe hlavičiek a aktuálneho stavu databázy zvolí či daný email prepustí ďalej alebo si vyžiada od užívateľskú interakciu, aby sa presvedčil o tom, že na druhej strane je človek. Systém môže v určitých prípadoch účinne blokovať všetkých užívateľov, ktorý nepoznajú adresáta osobne (aj v prípade, že nemajú v úmyslu užívateľa nejakým spôsobom poškodzovať/obmedzovať).

Všetko je to možné vďaka užívateľským databázam otázok, ktoré si definujú užívatelia sami. Tie sa neskôr využívajú k overovaniu či je odosielateľ človek, a to formou otázky, na ktorú je povinný odpovedať ak chce, aby sa jeho emaily doručili. Otázka je odoslaná odosielateľovi po

obdržaní emailu poštovým serverom a zistení, že kombinácia odosielateľ:prijímateľ nie je zatiaľ povolená. S obdržaním emailom sa naloží v závislosti od konfigurácie.

Môže sa buď to odložiť na disk alebo doručiť do schránky adresátovi, ale s pozmeneným Predmetom správy, do ktorého bude pridaný prefix (prednastavený je [UNVERIFIED]). Server si zapamätá túto konverzáciu a aj fakt, že odoslal otázku. Príklad takejto komunikácii môžete vidieť na obrázku 5.4. Popíšeme si rovno ako takáto operácia prebieha (predpokladáme, že Bob má nastavenú aspoň jednu otázku a Alice píše Bobovi po prvý krát a nie je uvedená na žiadnom whiteliste):

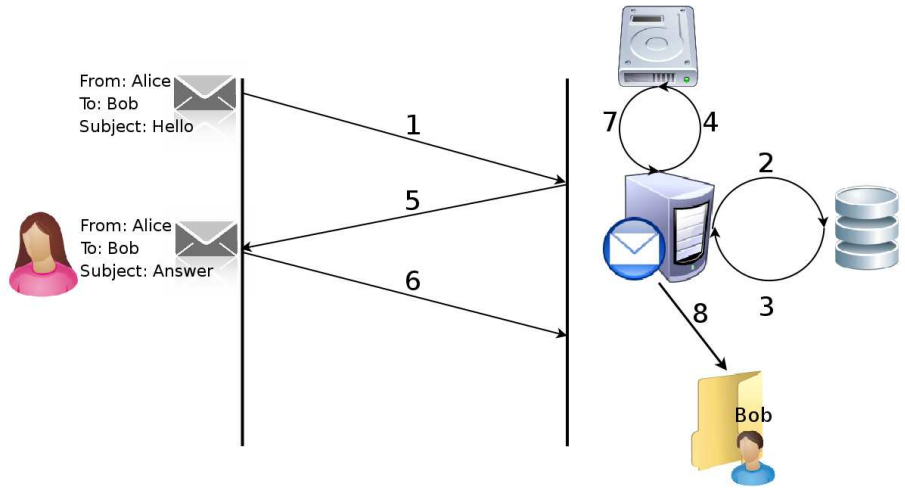


Obr. 5.4: Spampuzzle s Tag and Ask

1. Alica odošle Bobovi regulárny email
2. Poštový server Boba skontroluje či Alica nie je implicitne povolená prípadne či už správne nezodpovedala otázku
3. Keďže nenastala ani jedna situácia z predchodzieho bodu tak je v užívateľskej databázy Boba náhodne vylosovaná otázka
4. Predmet Alicinho emailu sa označí (otaguje) prefixom [UNVERIFIED] a doručí sa Bobovi do schránky
5. Na základe vylosovanej otázky sa zašle email s otázkou na adresu Alice
6. Alica odošle email so správnou odpoveďou, čo ju zaradí do databáze a už žiaden email od nej poslaný na Bobovu adresu nebude obsahovať značku UNVERIFIED

Ak Alica odpovie nesprávne, zašle sa v závislosti na uplynutom čase medzi odoslaním a odpoveďou opätovne otázka.

V konfigurácií zadrž (HOLD) email, ktorú vidíme na obrázku 5.5 je postup o niečo iný (opäť ako v predchádzajúcom príklade máme rovnaké podmienky):



Obr. 5.5: Spampuzzle s Hold and Ask

1. Alica odošle Bobovi regulárny email
2. Poštový server Boba skontroluje či Alica nie je implicitne povolená prípadne či už správne nezodpovedala otázku
3. Keďže nenastala ani jedna situácia z predchodzieho bodu tak je v užívateľskej databázy Boba náhodne vylosovaná otázka
4. Alicin email sa odloží na disk
5. Na základe vylosovanej otázky sa zašle email s otázkou na adresu Alice
6. Alica odošle email so správnou odpoveďou, čo ju povolí do budúcna odosielať emaily Bobovi
7. Vyzdvihne sa email z disku (prípadne všetky ostatné emaily, ktoré sa tam medzitým ukladali od Alice)
8. Email(y) sa doručia Bobovi

Pre ukážku si ešte ukážeme ako môže vyzerať email s otázkou. Databáza otázok mala takéto nastavenie

```
SK: Z ktorej krajiny pochádzam?
CZ: Z které krajiny pocházim?
EN: Where I came from?
==
Slovenská republika
Slovensko
Slovakia
Slovak Republic
\%\%}
```

Email s otázkou bude vyzerať takto

```
from alice@example.net
to bob@example.net
date Fri, Aug 1, 2008 at 6:32 PM
subject [Spampuzzle] Challenge response needed to verify you

SK: Z ktorej krajiny pochádzam?
CZ: Z které krajiny pocházim?
EN: Where I came from?
```

5.4 Inštalácia

Program je šírení pod licenciou GNU/GPL a obsahuje serverový perlovský skript, klientský BASH skript a taktiež základnú dokumentáciu popisujúcu inštaláciu ako aj konfiguráciu.

5.4.1 Prerekvizity

Program bol testovaný na operačnom systému GNU/Linux na distribúcii Debian Etch. Jeho funkčnosť by mala byť po splnený prerekvizit zachovaná aj na ostatných distribúciách prípadne iných odrodách systémov *NIX vďaka multiplatformovosti interpreta Perl. Základné balíčky nutné pre beh Spampuzzle sú:

- >=Perl 5.6.0

  - Net::Server
  - Net::SMTP
  - Encode
  - MIME::QuotedPrint::Perl
  - MIME::Base64
  - Switch
  - IO::Multiplex
  - BerkeleyDB
- berkeleydb
- >=postfix 2.3
- netcat

Kvôli podpore BerkeleyDB musíme mať nainštalovanú nielen systémovú knižnicu berkeleydb, ale aj knižnicu BerkeleyDB pre Perl. Doporučuje sa nainštalovať a nakonfigurovať si postfix s podporou SASL, avšak to nie je nutné.

5.4.2 Postup pri inštalácii

Vytvoríme užívateľa, pod ktorým bude spustený démon Spampuzzle (Master) a ktorý bude vlastníkom všetkých zadržaných emailov ako aj databáz. Master nepotrebuje k svojmu fungovaniu práva administrátora a tak z bezpečnostného hľadiska nie je ani doporučované sa pokúšať o jeho beh pod kontom administrátora. Pre účely tohto návodu si vytvoríme užívateľa 'spampuzzle', ktorý bude mať nastavený domovský adresár /var/spampuzzle. Do tohto adresára sa ukladajú všetky dáta spojené s chodom programu ako zadržané emaily a databázy.

```
# adduser --home /var/spampuzzle spampuzzle
```

Program je distribuovaný spolu s dvoma súbormi, ktoré je potreba uložiť na systém, kde máme poštového démona postfix.

```
# cp spampuzzle-filter.pl /usr/bin
# cp spampuzzle-client.sh /usr/bin
```

Nastavíme príslušné oprávnenia pre tieto dva súbory, aby ich mohol spúšťať novo vytvorený užívateľ 'spampuzzle'.

```
# chmod +x /usr/bin/spampuzzle.pl
# chmod +x /usr/bin/spampuzzle-client.sh
# chown spampuzzle /usr/bin/spampuzzle.pl
# chown spampuzzle /usr/bin/spampuzzle-client.sh
```

K úspešnej integrácií programu spampuzzle do nášho poštového démona postfix musíme upraviť konfiguráciu postfixu. V prvom rade si definujeme filter obsahu, ktorý bude ukazovať na transportný bod 'filter'. Tento transportný bod si potom zadefinujeme v súboru master.cf ako unixový socket, ktorý bude email preposielať za pomoci interného príkazu pipe do programu definovanom v argumentu "argv=", kde definujeme aj interné hodnoty získavané z postfixu pomocou pipe a to sender, recipient, client_address a sasl_username. Hodnota 10 udáva maximálny počet súbežných inštancií pipe s týmto argumentom. Táto hodnota by mala postačovať väčšine serverov. Avšak ak váš poštový server je bombardovaný emailami tak doporučujem zvýšiť túto hodnotu napr. na 100.

Ako posledný krok k integrácií s postfixom je nutné nastaviť dodatočný transport, ktorý bude slúžiť ako ďalší SMTP démon, cez ktorý bude Master odosielať legitímnu poštu. Transport nesmie obsahovať spampuzzle ako filter obsahu, ináč by sa vytvoril cyklus v rámci odosielania emailov. Vo výsledku je nutné upraviť tieto dva súbory a pridať tam nasledujúce riadky

/etc/postfix/main.cf:

```
content_filter = filter:dummy
```

/etc/postfix/master.cf:

```
filter unix - n n - 10 pipe
user=spampuzzle argv=/usr/bin/spampuzzle-client.sh ${sender}
${recipient} ${client_address} ${sasl_username}

localhost:10025 inet n - n - 10 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 smtpd_error_sleep_time=0
-o smtpd_soft_error_limit=1001
-o smtpd_hard_error_limit=1000
-o smtpd_use_tls=no
```

Ak by ste chceli nastaviť iné porty pre transport bez filtru, tak je nutné nastaviť túto novú adresu v hlavičke spampuzzle-filter.pl skriptu a to v premennej $mailserver.

Zmena nastavení Postfixu sa prejaví až po vynútený opätovného načítania konfigurácie

```
postfix reload
```

Posledným krokom je spustenie nášho démona(Master) s patričnými parametrami (viď 5.5). Ak sa rozhodnete prevádzkovať Mastera na inom porte ako predvolenom musíte tento port zmeniť aj v súbore spampuzzle-client.sh a to zmenou hodnoty premennej SPAMPORT na vami zvolený port.

Proces inštalácie by mal byť v tomto bode ukončený. Funkčnosť programu si môžeme overiť zaslaním konfiguračného emailu alebo zvýšením výrečnosti nášho programu na najvyššiu úroveň (-v -v -v), zaslaním ľubovoľného emailu a sledovanie výpisov v záznamovom súboru (napr. /var/log/mail.log).

5.5 Konfigurácia

Ani jeden z modulov Spampuzzle neobsahuje konfiguračný súbor (výnimku tvoria súbory s obsahom whitelist IP a emailových adries). Všetky globálne nastavenia sa nastavujú v prípade Mastera pomocou predávania argumentov. Užívateľské nastavenia sa však ukladajú do databáze, takže sa o ne pri reštartoch neprichádza.

5.5.1 Konfigurácia Mastera

Master sa konfiguruje pomocou vstupných argumentov. Väčšina parametrov ma definovanú prednastavenú hodnotu, ktorá sa použije v prípade ak

parameter nebol zadaný. Zoznam argumentov a ich popis nájdete v tabuľke 5.2. Master nemôže byť súčastne spustený viac krát. Kontrola je zabezpečená cez zamykacie mechanizmy. V prípade nekorektného ukončenia démona je doporučené skontrolovať adresár /tmp a v ňom hľadať súbor spampuzzle.lock. Súbor symbolizuje bežiacu inštanciu, ak však žiadna nebeží tak tam nemá čo robiť a je treba ho zmazať, aby mohol byť démon opätovne spustený.

V prípade potreby je taktiež k dispozícií autentifikované spojenie Mastera na poštový server. Toto spojenie slúži k zasielaniu otázok, prípadne k finálnemu doručovaniu. Prihlasovacie meno a heslo sa definuje na začiatku súboru spampuzzle-filter.pl v premenných login a pass.

Užívateľské nastavenia sú uložené v databázy a preto nie je nutné mať pre nich konfiguračný súbor. Užívatelia si menia svoje nastavenia pomocou konfiguračných emailov (viď 5.5.2). Administrátor môže cez vstupné argumenty vynútiť prednastavenú hodnotu v prípade dvoch parametrov, ktoré môžu konfigurovať užívatelia (case a tag). Ostatné parametre nie sú konfigurovateľné užívateľom, nakoľko ich deaktivácia/aktivácia môže mať veľký vplyv na výkon a povahu Spampuzzle.

Master taktiež pracuje s dvojicou súborov, ktoré obsahujú zoznam IP adries a emailových adries. V prípade emailových adries tento zoznam slúži k povoleniu privilegovaných príkazov za pomoci konfiguračných emailov. Tieto príkazy by mal byť schopný spúšťať každý užívateľ na poštovom serveru, ale nevylučuje sa možnosť kde by mu boli otázky nastavené automaticky, prípadne, by si ich mal možnosť nastaviť iba po určitú dobu (napr. týždeň po vytvorení konta). Ak váš poštový server podporuje SASL, a užívatelia majú vytvorené patričné kontá, tak tento whitelistovací mechanizmus IP adries je nepotrebný. IP adresy môžu byť v tomto zoznam zapísané v klasickom zápise, CIDR zápise a aj ako regulárny výraz. Druhý zoznam obsahuje emailové adresy, na ktoré nebude zasielaná žiadna otázka, a ktoré budú automaticky povolené.

Oba súbory sa načítajú pri štarte Mastera a pri zmene týchto súborov je treba vynútiť znovu načítanie a to poslaním signálu HUP Masterovi.

Tabuľka 5.2: Argumenty démona Spampuzzle

Argument	**Popis**
-h, --help	Výpis povolených argumentov
--version	Výpis verzie programu

Pokračovanie na ďalšej stránke ...

Tabuľka 5.2 Argumenty démona Spampuzzle

Argument	Popis
--man	Manuálová stránka v perl-doc (argument je závislý na funkcii z balíčku perl-doc, ktorý musí byť nainštalovaný)
-v, --verbose	Zvýši stupeň podrobnosti záznamov čo ma za následok podrobnejšie výpisy programu. Argument je možno definovať viac krát a tým ďalej zvyšovať jeho stupeň. Maximálny počet opakovaný je 3.
-q, --quiet	Zníži stupeň podrobnosti záznamov
-i, --inet=[HOST:]PORT	Nastavenie IP adresy HOST a portu PORT na ktorom bude démon počúvať. Ak nie je zadefinovaná adresa, tak sa implicitne použije localhost (127.0.0.1). Prednastavený port je 1025
-d, --daemonize	Spustí program, vytvorí démona a vráti sa do shellu.
--pidfile=PATH	Uloží process ID (PID) démona do súboru PATH
--user=USER	Spustí démona pod užívateľom USER (default: spampuzzle)
--group=GROUP	Spustí démona pod skupinou GROUP (default: nogroup)
--dbdir=PATH	Umiestni databázové súbory do PATH (default: /var/spampuzzle)
--hostname=NAME	Nastaví hostname na NAME (default: ‘hostname‘)
--whitelist-sender=FILE	Zoznam povolených odosielateľov (default: /var/spampuzzle/whitelist.db)
--whitelist-ip=FILE	Zoznam povolených IP adries(regexp a CIDR je tiež povolený), ktoré budú môcť využívať konfiguračné emaily (default: /var/spampuzzle/whitelist_ip.db)
--listen-queue-size=N	Maximálne N spojení môže čakať súbežne na socketu
-m --drop-spam	Zahodí emaily označene v hlavičke ako SPAM
-a --ack-verify	Zašle oznámenie o úspešnej verifikácií

Pokračovanie na ďalšej stránke ...

Tabuľka 5.2 Argumenty démona Spampuzzle

Argument	Popis
-s --strict-sasl	Kontrola zhody SASL užívateľského mena a adresy odosielateľa. Ak sa nenájde zhoda, tak je takáto relácia považovaná za neautentizovanú a tak je aj s ňou nakladané
--subject-only	Kontrola iba predmetu na zhodu s očakávanou odpoveďou
-t --tag	Označí predmet emailu prefixom [UNVERIFIED] a doručí ho. Zadržiavanie emailov, je teda deaktivované. Užívateľ si však môže vynútiť opäť zadržiavanie pomocou konfiguračného emailu.
--max-retry=N	Maximálny počet zaslaní otázky jednému užívateľovi. Po dosiahnutí tohto limitu sa prestane démon pokúšať o opätovné zaslanie otázky (default: 3)
--retry-delay=N	Minimálny počet sekúnd medzi jednotlivými pokusmi o opätovné doručenie otázky. Opätovné doručenie je vyvolané obdržaným emailom, ktorý neobsahuje správnu odpoveď (default: 3600 sekúnd)
-w --whole-message	Vynúti kontrolu celého emailu a nie len prvého riadku obsahujúceho "nebiely" znak

5.5.2 Konfiguračné emaily

Po inštalácii emaily prechádzajú cez nášho Slavea do Mastera a od neho späť do druhého poštového démona, bez akejkoľvek úpravy. Aby spampuzzle niečo robil, tak potrebuje mať k dispozícii užívateľské otázky. Bez toho, aby si ich užívateľ nastavil bude spampuzzle iba prietokový ohrievač, ktorý však nebude ohrievať.

Jedinou cestou ako si nastaviť tieto otázky, prípadne meniť užívateľské nastavenia je pomocou konfiguračných emailov obsahujúcich text vo formáte spampuzzle. Konfiguračný email je charakterizovaný tým, že položky "odosielateľ" a "prijímateľ" obsahujú tie isté adresy a zároveň je zhodná s užívateľským menom SASL. V prípade ak nie je nastavený SASL tak musí byť email poslaný z jednej z IP adries špecifikovaných

v whitelistip súboru. Ak nie je, tak je tento email považovaný za bežnú komunikáciu.

Splnením týchto požiadaviek môžeme využívať výhod konfiguračných emailov. Predmet takýchto emailov sa kontroluje na výskyt riadiacich príkazov, ktoré menia význam emailu. Konfiguračný email je po spracovaní zahodený a nikde sa nearchivuje.

Zoznam riadiacich príkazov

- add

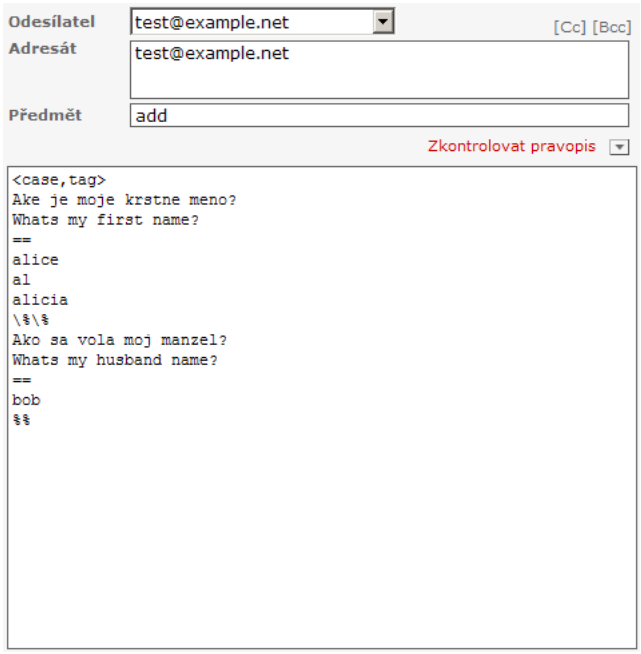
  Pridávanie užívateľských otázok. Telo emailu obsahuje otázky spolu s odpoveďami. Na jednu otázku je možné definovať viacero odpovedí. Otázka môže byť aj viac riadková a tým sa zjednoduchšuje možnosť zadávania komplexnejších otázok, prípadne viacjazyčných otázok. Otázky a odpovede sú oddelené riadkom obsahujúcim iba reťazec "==". Pri viacerých možnostiach odpovede je každá z týchto odpovedí na samostatnom riadku. Sekciu odpovedí ukončujeme riadkom obsahujúcim iba znaky "%%".

  V tomto emaily môžeme zároveň meniť aj naše užívateľské nastavenia a tým zmeniť chovanie filtru. Užívateľ si môže nastaviť či chce, aby sa v odpovediach rozlišovali veľkosti písmen alebo či sa majú emaily zadržiavať či označovať predmet prefixom. Prvý riadok slúži práve pre tieto nastavenia. Ak neobsahuje špeciálne formátovaný reťazec, tak bude považovaný za súčasť otázky, avšak ak obsahuje reťazec v tvare $< STR, STR >$, tak je braný ako konfiguračný. Existujú 4 možnosti zápisu tohto prvého riadku.

  - <case,tag> nastaví rozlišovanie písmen a označovanie predmetu
  - <,tag> nastaví nerozlišovanie písmen a označovanie predmetu
  - <case,> nastaví rozlišovanie písmen a zadržiavanie emailov
  - <,> nastaví nerozlišovanie písmen a zadržiavanie emailov

  Na tento email systém nezasiela žiadnu odpoveď o úspechu alebo neúspechu. Otázky v sebe nesmú obsahovať reťazec "==". Odpovede nesmú obsahovať znak '|' a taktiež reťazec "%%". Pri zmene konfigurácie z untagged na tagged doručovanie sa emaily predtým zadržané nedoručia automaticky, ale až po príslušnom povolený za pomoci wh alebo zodpovedania otázky.

K účelom overenia databázy otázok a nastavení slúži príkaz *qlist*. Každý ďalší 'add' email bude pripájať otázky na koniec užívateľského zoznamu otázok. Ukážkový email, ktorý pridáva dva otázky a nastavuje rozlišovanie písmen a označovanie predmetu môžete vidieť na obrázku 5.6.

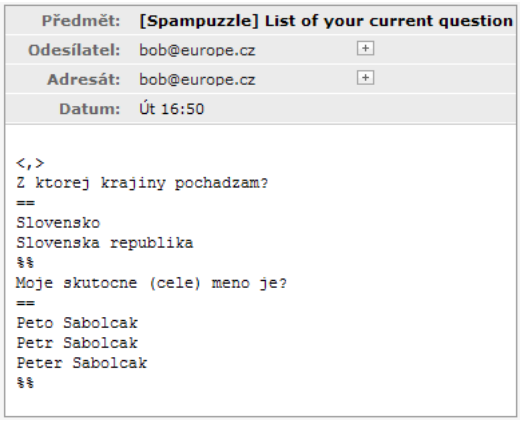


Obr. 5.6: Ukážkové použitie add

- qlist

  Slúži k získaniu zoznamu otázok a odpovedí ako aj aktuálnych nastavení. Telo správy je v tomto prípade ignorované a tak jedine čo stačí nastaviť je predmet správy. Po odoslaní emailu s predmetom 'qlist', bude obratom zaslaná odpoveď, ktorá bude obsahovať zoznam otázok a taktiež nastavenia. Ak užívateľ nemá zadané žiadne otázky, tak mu bude zaslaný email s informáciou o tejto skutočnosti. V opačnom prípade užívateľ obdrží email, v ktorom budú na prvom riadku nastavenia (case, tag) a v ostatných budú otázky a odpovede. Celá správa je vo formáte, ktorý je popísaný v príkazu add. Je to hlavne z dôvodu ľahšieho vytvárania záloh otázok a nastavení a ich obnovy. Tu využijeme napríklad pri pokuse o zmenu

niektorej z otázok prípadne odpovedí. Editácia otázok nie je možná a tak neostáva nič iné len zmazať celý obsah užívateľskej databázy (delete_all) a nahrať späť pozmenený obsah pomocou príkazu add. Otázky sú zobrazené v opačnom poradí v akom boli pridávané, tzn. hornej časti emailu sa nachádzajú najnovšie otázky a v dolnej najstaršie. Ukážkový výstupný email môžete vidieť na obrázku 5.7.



Obr. 5.7: Výstup príkazu qlist

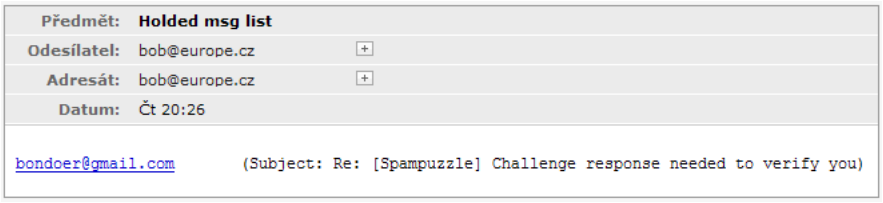
- delete_all

  Výmaz celého obsahu užívateľskej databáze, zahrňujúc otázky a odpovede ako aj užívateľské nastavenia. Telo tohto emailu je ignorované. Vďaka výmazu otázok sa zároveň deaktivuje spampuzzle pre daného užívateľa, nakoľko sa v databáze nenachádzajú žiadne otázky, ktoré by mohol démon posielať. Telo tohto emailu je ignorované. Pozor, v prípade, že sa zmaže databáza otázok a opäť nahrá (v zmenenom poradí prípadne úplne iný obsah otázok), mechanizmus nemá možnosť detekovať tieto zmeny. Takže môže nastať situácia keď niekto dostane otázku pred takouto operáciou, ale zodpovie je ju až po tom ako si zmeníme databázu. V tomto prípade ak pôvodná otázka mala identifikačné číslo väčšie ako je aktuálny maximálny počet otázok, tak sa vyberie opäť náhodne nová otázka a odošle sa. Mimo to sa resetuje indikátor počtu zaslaných otázok na 1. Ak sa však číslo otázky prekrýva so súčasným nastavením otázok, tak systém na túto zmenu nebude reagovať a bude sa správať ako za bežných podmienok, tj. zistí či v emailu bola nájdená správna

odpoveď a ak nie tak mu za vhodných podmienok (po uplynutý čakacej doby medzi zasielaním otázok) zaslaná *nová* otázka. Preto je dôležité pred výmazom databázy skontrolovať zoznam blokovaných emailov (hlist) a až po patričnom povolení za pomoci príkazu wh vymazať našu databázu.

- hlist

  Po odoslaní emailu s týmto predmetom (telo emailu je ignorované) je obratom doručený zoznam blokovaných emailov. Zoznam obsahuje email odosielateľa a predmet správy. Pre odblokovanie emailov stačí použiť príkaz wh. Ukážkový výstupný email môžete vidieť na obrázku 5.8. V niektorých prípadoch môže byť aj po zaslaní wh príkazu zablokovaný email. V tomto prípade, je poškodená databáza a je treba kontaktovať administrátora (email je stále k dispozícii na disku, len podľa databázy nie je zadržaný).

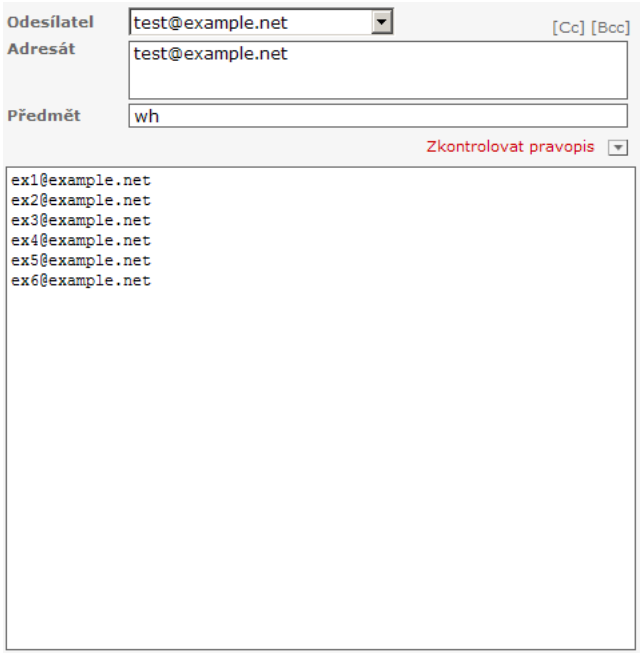


Obr. 5.8: Výstup príkazu hlist

- wh

  Telo tohto typu správy obsahuje zoznam emailových adries, ktoré budú povolené bez nutnosti zodpovedať otázku. Príkaz môže odblokovať prípadný zablokovaný email a povoliť ďalšie emaily od tohto užívateľa. Zoznam môže obsahovať viacero emailových adries. Čo riadok, to adresa. Riadok sa však v prípade zadania nekorektnej adresy bude ignorovať. Adresa je korektná ak obsahuje . Príklad takéhoto emailu môžete vidieť na obrázku 5.9.

- bh



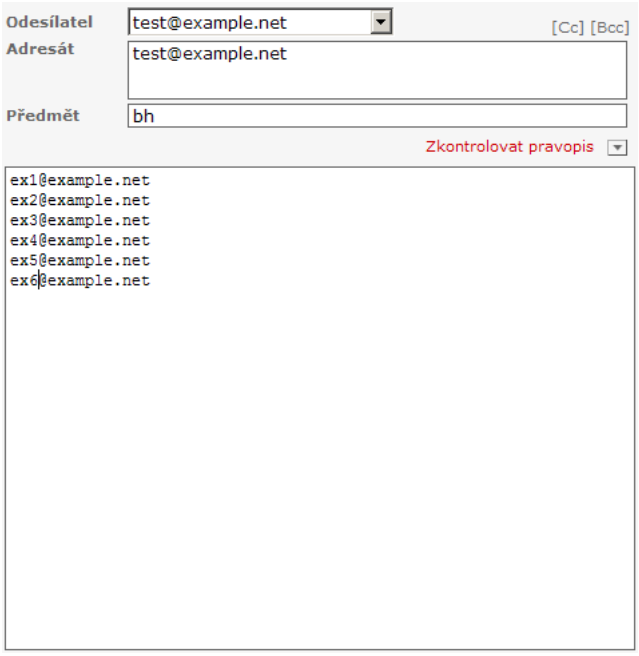
Obr. 5.9: Ukážkové použitie príkazu wh

Telo tohto typu správy obsahuje zoznam emailových adries, ktoré budú vymazané z užívateľskej databáze povolených emailových adries. Toto nastavenie nemá nič spoločné s adresami načítaními pri spúšťaní programu z hodnoty argumentu whitelist-sender. whitelist-sender má vždy väčšiu hodnotu ako užívateľské nastavenia. Po nastavení tohto parametru, bude každý ďalší email z definovaných adries opäť dotázaný o zodpovedanie otázky. Ako v prípade wh aj pri tomto príkazu je možné zadefinovať viacero emailových adries a to rovnakým spôsobom (čo riadok, to adresa). Príklad takéhoto emailu môžete vidieť na obrázku 5.10.

5.5.3 Whitelist

Whitelist-sender Už na začiatku používania je dobré si nastaviť naše známe adresy, domény alebo užívateľov, ktorých poznáme a teda nechceme, aby sa museli obťažovať zadávaným odpovede na našu otázku. Pre tento účel tu je možnosť definovať takéto kritéria v súbore. Cestu k tomuto súboru definujeme cez vstupné argumenty démona Spampuzzle (viď tabuľku 5.2).

Tento súbor môže obsahovať rôzne zápisy emailových adries a preto



Obr. 5.10: Ukážkové použitie príkazu bh

si ich rozoberieme.

- **/regexp/**
  Email odosielateľa sa bude kontrolovať voči tomuto regulárnemu zápisu, a tak je možné vytvoriť takmer ľubovoľnú množinu týchto zápisov, ktoré budú rozpoznávať veľký okruh rôznorodých adries. Zápis v plnej miere dokáže nahradiť všetky ostatné zápisy. Avšak narozdiel od nich je pomalší pri vyhodnocovaní a v rámci optimalizácie by sa mal tento zápis využívať v rozumnej miere a skôr uprednostňovať ostatné formy zápisu.

- **user@domain**
  Exaktný zápis emailovej adresy.

- **user@**
  Zápis, ktorý sa zhodne z každou emailovou adresou, ktorá bude od užívateľa user, z ľubovolnej domény.

- **domain**
  Zápis, ktorý sa zhodne z každou emailovou adresou, ktorá bude z domény domain, od ľubovolného užívateľa.

Whitelist-IP Ďalším zo súborov, ktorý obsahuje whitelistovú databázu je súbor obsahujúci IP adresy privilegovaných užívateľov, ktorý ak sú uvedený v tomto zoznamu, sú schopný bez SASL autentifikácie pracovať s konfiguračnými emailami a takisto autowhitelistovať svoju emailovú komunikáciu. IP adresy môžu byť zadávané v klasickom formáte, v CIDR alebo za pomoci regulárnych výrazov.
Konverzácia sa teda môže povoliť 4 spôsobmi:

- whitelist-sender
- SASL autentifikovaným odosielaným emailom cez náš SMTP server
- povolený privilegovaný rozsah pomocou whitelist-ip a následným odosielaním emailov cez náš SMTP server
- poslaním konfiguračného emailu s predmetom 'wh'

Kapitola 6

Programátorská príručka

6.1 Perl

Perl je interpretovaný jazyk, ktorý je vhodný na riešenie problémov týkajúcich sa spracovania reťazcov a vďaka jeho širokej podpore umožňuje tieto problémy riešiť na širokej škále platforiem. Medzi jeho ďalšie výhody nepochybne patrí aj dostupnosť viac ako 4000 modulov, ktoré implementujú nespočet rozhraní, za pomoci ktorých môžete odosielať email cez SMTP server, pripájať sa na relačnú databázu alebo vytvoriť si vlastného démona. Nesmieme zabudnúť na dynamickú prácu s pamäťou a rýchlosť práce s regulárnymi výrazmi.

Za tieto výhody však musíme zaplatiť určitú cenu. Tá spočíva vo väčšej spotrebe pamäte a tak v rámci projektu spampuzzle boli využité optimalizačné techniky, ktoré by mali zabraňovať tomuto javu pri väčších záťažách. V prvom rade sa žiadnom spôsobom nemodifikoval stack príkazom **shift**. Miesto tohto príkazu sa pracovalo so zoznamom vstupných argumentov procedúr, ktorý je reprezentovaný premennou `@_`.

V prípade premenných, ktoré obsahujú obsah emailu, ktorý môže byť veľký aj niekoľko megabyteov bolo minimalizované ich kopírovanie. Uprednostňovali sa odkazy priamo na vstupný zoznam argumentov `@_`.

Regulárne výrazy boli pre zvýšenie rýchlosti, v niektorých prípadoch predkompilované.

Vďaka týmto opatreniam pamäťová náročnosť programu klesla v niektorých prípadoch o rovných 50%! Napríklad pri vykonávaní hodnota spotrebovanej pamäte pred optimalizáciou bola 80 MB a po nej 40 MB.

6.1.1 Modul Net::Server::Multpiplex

Modul je navrhnutý k práci z viacerými spojeniami v rámci jedného procesu. Je postavený nad knižnicami Net::Server a IO::Multiplex, ktoré kombinuje do ľahko použiteľného rozhrania.

Bohužiaľ knižnica neovplýva výrečnosťou v dokumentácií a zároveň nie je obsiahnutá v mnoho projektoch, z ktorých by sa dali čerpať ukážkové informácie o jej používaní. Ak sa však bude používať pre základné úkony je skutočne ľahko použiteľná. Naproti tomu pri potrebe práce s premennými v rámci objektu nestačí mať základné znalosti o Perlu a moduloch, ktoré využíva.

6.2 Databáza

K uchovávaniu údajov o užívateľoch ako aj o zrealizovaných rozborov kombinácií odosielateľ:prijímateľ sa využívajú embedded databáza a to konkrétne BerkeleyDB. Pre tento projekt bola vybraná vďaka svojej popularite a rýchlosti.

Embedded databáza obsahuje iba dva atribúty; kľúč a jeho hodnotu. Vzhľadom k tomu, že je možné vyhľadávať iba podľa kľúča, tak voľba kľúča je zásadná.

Spampuzzle obsahuje celkovo dva databázové súbory. Prvý uchováva užívateľské nastavenia a otázky. V druhom sa uchovávajú záznamy emailových komunikácií, ktoré analyzoval spampuzzle.

6.2.1 Databáza užívateľských otázok a nastavení

Užívateľská databáza je uchovaná v súboru spampuzzle_qa.db, ktorý je umiestnený v adresári definovanom argumentom dbdir. Ak argument nie je definovaný, tak sa nachádza v prednastavenom adresári /var/spampuzzle. V tomto súboru sú uložené užívateľské nastavenia pre všetkých užívateľov.

V databázy môžeme vyhľadávať v závislosti podľa typu informácie, ktorú chceme získať podľa dvoch typov/formátov kľúča.

Prvý typ je identifikovaný emailovou adresou užívateľa. Hodnota týchto typov kľúčov v sebe ukrýva základné údaje platné pre daného užívateľa

$$email@europe.eu => N;N;N$$

- nastavenie (ne)rozlišovania veľkosti písma v odpovediach

Kľúč	**Hodnota**
user@example.net	1;1;2
user@example.net;1	Ako sa volam\nWhat's my name ?==Peter%%Peto
user@example.net;2	Kde zijem?\nWhere do I live ?==Czech republic%%Ceska republika

Tabuľka 6.1: Príklad užívateľskej databázy

- nastavenie označovania nedôveryhodných emailov prefixom v predmete alebo zadržiavanie emailov
- počet zadefinovaných otázok tohto užívateľa

Hodnota tohto kľúča sa nastavuje za pomoci konfiguračných emailov, ktoré obsahujú v predmete slovo add. Všetky tieto tri hodnoty sú uložené v jednom poli a tak v rámci tohto poľa sú oddelené oddeľovačom ';'. Prvé dve hodnoty obsahujú nastavenie rozlišovania písmen a označovania emailov. Môžu nadobúdať hodnoty 0,1. 0 v prípade rozlišovania písmen znamená, že sa nebude požadovať pri kontrole odpovede zhoda veľkosti písmen zatiaľ čo 1 bude vyžadovať túto zhodu. V prípade označovania emailov znamená 0, že sa takéto emaily nebudú označovať a miesto toho budú odkladané na disk. Číslo 1 na tomto mieste určí, že sa predmet emailu označkuje prefixom a doručí pôvodnému adresátovi.

Posledná položka tohto poľa je dôležitá pre tvorbu druhého typu kľúča. Ten je tvorený dvojicou emailová adresa (totožná z predchádzajúceho kľúča) a identifikačné číslo otázky. Práve celkový počet zadefinovaných otázok nám určí rozsah, z ktorého si môže program vyberať. Číslovanie otázok začína číslom 1. Hodnota, ktorá je pridelená ku kľúču obsahuje otázku a odpovede. Od seba sú oddelené oddeľovačom '=='. Oddeľovač '%%' sa používa v situáciách keď je definovaných viacero odpovedí.

Ukážkový záznam tejto databázy je v tabuľke 6.1

6.2.2 Komunikačná databáza

V druhom typu databázy je uchovaná každá komunikácia, ktorá prešla cez náš systém a nespadala do kategórie whitelist-sender (viď 5.5.3). Súbor, v ktorom je uložená databáza sa volá spampuzzle.db. Tak isto ako v prípade užívateľskej databáze, aj táto sa nachádza v ceste definovanej argumentom dbdir prípadne v prednastavenej ceste /var/spampuzzle.

Kľúč	**Hodnota**
user@universe.net;user@example.net	1|1|1|20080801232759|124f26a0807311306n1b-fbc915oeaf9786b69d00b15@mail.gmail.com
user@europe.eu;user@example.net	2||||

Tabuľka 6.2: Príklad komunikačnej databázy

Komunikácia je identifikovaná kombináciou odosielateľ, prijímateľ. Táto kombinácia zároveň slúži ako kľúč, podľa ktorého sa vyhľadávajú záznamy v komunikačnej databáze. Medzi kombináciu je vložený oddeľovač ';', aby sa predišlo prípadným konfliktom a zabezpečila sa jednoduchá konštrukcia kľúča. Pole identifikované týmto kľúčom v sebe obsahuje tieto informácie:

- status komunikácie, ktorý môže nadobúdať dvoch hodnôt
  - 1 verifikačný email s otázkou bol zaslaný a čaká sa na odpoveď
  - 2 úspešne autorizovaná komunikácia
- počet zaslaných verifikačných emailov. Hodnota sa využíva pri zisťovaní či nebol dosiahnutý maximálny počet opakovaných zaslaní verifikačného emailu pre túto komunikáciu.
- identifikačné číslo otázky, ktorá bola zaslaná vo verifikačnom emailu. Ak budú opätovne zaslané verifikačné emaily, tak budú obsahovať túto otázku (ak bude k dispozícii). Mechanizmus slúži na ochranu pred získaním našej databázy otázok. Vďaka tomuto číslu taktiež vieme aká má byť správna odpoveď na otázku.
- časové razítko určujúce kedy bol odoslaný posledný verifikačný email. Opätovné zasielanie verifikačného emailu môže byť zaslané až po určitej dobe, ktorá sa počíta z tejto hodnoty a aktuálneho času. Ak ich rozdiel je väčší ako zadefinovaná hodnota v argumentu max-delay, tak verifikačný email môže byť odoslaný.
- zoznam zadržiavaných emailov. V zozname sú emaily identifikované svojím identifikačným reťazcom, ktorý im pridelil poštový server. Identifikačné reťazce sú oddelené znakom ','.

email_from@europe.eu;email_to@universe.un =>
N|N|N|YYYYMMDDHHmmss|MSGID,MSGID

Ako už je naznačené príkladom v tabuľke 6.2, je možné mať položku, ktorá bude mať vyplnený iba status a to hodnotou 2. Tento status symbolizuje, že je konverzácia povolená a preto sú ostatné atribúty nepodstatné a v rámci povoľovania sú odstraňované.

Pri práci so zadržanými emailami sa môžu užívatelia dostať do situácie, v ktorej nebudú môcť povoliť určitú komunikáciu. Tá bude povolená, ale v zozname, vygenerovanom príkazom hlist, im stále budú figurovať emaily od týchto povolených ľudí. Táto situácia znamená, že dáta na disku a v databáze nie sú zosynchronizované. Keďže sa vždy prv ukladá email na disk až potom sa zapisujú informácie o ňom do databázy, tak je vo väčšine prípadoch problém práve v databáze. V takýchto prípadoch musíme ručne doručiť email (podľa adries zistíme zložku a v nej sú už emaily uložene bezo zmeny) do cieľovej schránky a to buď cez mechanizmus príkazu sendmail alebo podobných. Databázu overíme za pomoci nástroj BerkeleyDB ako db_dump a pokúsime sa lokalizovať databázový problém.

6.3 Spracovanie emailu

Vstupné rozhranie démona spampuzzle je reprezentované z programového hľadiska funkciou

mux_eof ($mux,$fh,$input)

Funkcia je obsiahnutá v Perlovskej knižnici Net::Server::Multiplex, ktorá ju volá v okamihu keď klient skončil zápis. V premennej *input* je uložený celkový obsah emailu, ktorý nám zaslal klient. Vzhľadom k vlastnosti knižnici Net::Server::Multiplex sme museli zadefinovať funkciu mux_input(), ktorá však v našom programu nemôže byť použitá, keďže potrebujeme obdržať celý email, ktorý buď to uložíme alebo posielame ďalej.

Ako prvé musíme z vstupu odstrániť našu hlavičku označenú prefixom X-Legit. Vďaka hlavičke sme schopný zistiť odosielateľa a prijímateľa (v rámci hlavičky v tele emailu môže byť sfalšovaný) a taktiež prípadne SASL užívateľské meno.

6.3.1 Legitímny Email

Ak sa v hlavičke nachádzalo SASL meno a zhodovalo sa s emailovou adresou odosielateľa (v prípade zapnutej voľby strict-sasl), tak je email považovaný za legitimný. Za legitímny sa považuje aj v prípade ak bol

poslaný z whitelist-ip adresy. Legitímny email môže byť vyhodnotený dvoma spôsobmi:

- Odosielateľ a príjemca je rovnaký. Email je považovaný za konfiguračný a hľadá sa zhoda medzi konfiguračnými príkazmi a predmetom. Keď sa zhodujú, tak sa vykoná patričná operácia, v opačnom prípade je email doručený.
- Odosielateľ a príjemca nie je rovnaký. Komunikácia bude vzhľadom ku skutočnosti, že pochádza od autentifikovaného užívateľa povolená a zapísaná do komunikačnej databázy. Emaily, ktoré boli v rámci tejto komunikácie podržané budú po tejto operácií uvoľnené a doručené.

6.3.2 Regulárny email

Regulárny email je každý email, ktorého odosielateľ nebol autentifikovaný našim poštovým serverom jednou z metód SASL mechanizmu alebo nebol poslaný z privilegovaného rozsahu IP adries.

Ako prvé položíme dotaz databáze, či sa takáto konverzácia už neobjavila.

Pre každú novú konverzáciu vytvoríme patriční záznam v databáze a vložíme doňho všetky potrebné dáta. V závislosti od nastavenia (tag), buď sa email uloží na disk do adresára špecifikovaného argumentom holddir (prednastavený je /var/spampuzzle/hold) alebo sa pozmení predmet (pridá sa prefix [UNVERIFIED]) a doručí sa pôvodnému adresátovi. Ak má byť email uložený na disk, tak sa spravidla ukladá do súboru /holddir/prijemca/odosielatel/msgid. Po týchto krokoch sa pozrieme do užívateľskej databázy, kde ako prvotný kľúč bude vystupovať adresa prijímateľa. Zistíme počet otázok a tým môžeme vybrať náhodnú hodnotu z rozsahu 1..<počet otázok>. Vytvoríme druhotný kľúč (prijímateľ číslo otázky), za pomoci ktorého získame po rozparsovaní otázku a príslušne odpovede. Otázka je potom zaslaná pôvodnému odosielateľovi a do databázy je táto konverzácia uchovaná so statusom 1 a ostatnými patričnými údajmi.

V prípade opakovaného výskytu konverzácii overujeme jej status v databázy. Keď je rovný 2, tak email doručíme. V opačnom prípade musí byť rovný 1, to znamená, že čakáme na odpoveď, ktorá sa môže skrývať práve v tomto emailu. Kontrola sa pokúša nájsť zhodu s odpoveďou, ktorá bola uložená v komunikačnej databázy a to v predmetu a ak cez vstupné argumenty programu nie je zakázaná kontrola tela (subject-only), tak aj v tele emailu.

Po nájdení správnej odpovede zmeníme status konverzácie na 2 (povolíme ju) a odblokujeme prípadné zablokované emaily. V opačnom prípade sa overí čas posledného zaslania otázky a aktuálny čas a ak ich rozdiel bude väčší ako čas definovaný v argumente max-delay a či nebol prekročený limit na počet opakovaných zaslaní otázky. Po splnení požiadaviek pre opätovné zaslanie otázky, sa otázka načíta z užívateľskej databázy a odošle sa. Otázka v rámci konverzácie je vždy rovnaká.

Kapitola 7

Testy

Spampuzzle pri svojej práci vykonáva porovnávania obsahu emailov s databázovými údajmi. To sú samozrejme náročné operácie a hlavne ak v takýchto situáciách figuruje práca s diskom. Prístupy na disk sú pri práci s databázou a pri ukladaní/vyzdvihovaní zablokovaných emailov. Tieto operácie sú nosnou líniou, na ktorú sa tieto testy zamerali.

7.1 Testovacie prostredie

K testovaniu bol použitý poštový server s nasledujúcou hardwarovou špecifikáciou:

Základná doska	Intel S5000VSA
Procesor	2 x Xeon E5310@1.60GHz
Pamäť	4 x 1 GB DDR2 667MHz ECC
Disk	2 x SATA 500GB Western Digital WD5000ABYS

Ako operačný systém bol použitý Debian GNU/Linux 4.0. Ako filesystém bol zvolený Ext3 a nad nim nakonfigurovaný softwarový RAID 1. Server bol pripojený rýchlosťou 100 MBit/s full duplex.

7.2 Metódy

Emaily odosielané na testovacie konto boli vygenerované program smtp-source. Ten je dodávaný v rámci Postfixu ako testovací nástroj SMTP. Pomocou neho si môžeme vygenerovať potrebnú záťaž na poštovom serveru. My sme ho použili v tejto konfigurácii

smtp-source -m 100 -c -s 10 -l 1000000 -t test@hammer.cz hammer.cz

kde -m udáva počet správ odoslaných týmto programom, -s počet konkurentných spojení na server a -l je veľkosť tela emailu. Pomocou tohto programu a sledovania výpisov na cieľovom serveru (hammer.cz), sme zaznamenávali časy jednotlivých testov. smtp-source posiela celý balík okamžite a za použitia všetkých spojení (ak je to možné). Avšak ak sme dali viac spojení ako 10 potýkali sme sa s preťažovaním serveru. Nepomohlo ani zvyšovanie maxproc parametru v master.cf. Testy aj napriek tomu boli vykonané pri nastavení 100 maxproc pre náš klientský skript.

Zvolili sme si dva druhy testovaní. Jedno pokrýva štandardnú emailovú komunikáciu, ktorá sa pohybuje v medziach 10 kB (test "Malý email") a druhé pokrýva záťažový test pri prijímaní emailov o veľkosti 1MB (test "Veľký email"). Spampuzzle bol spustený s defaultnými nastaveniami a v rámci neho nič nebolo zoptimalizované.

7.3 Merania

Merania pri vyšších počtoch emailov už nedokázal server zvládať a tak posielal chyby typu Temporary table lookup failure. Pre odľahčenie dotazov som použil proxy: parameter, ktorý ani pri obrovských záťažiach nepomohol. Preto sa pre účely testu robil vytvorila databáza BerkeleyDB, obsahujúca jedného (testovacieho) užívateľa, ktorému sa posielali emaily. Tým sme zmaximalizovali výkon nášho servera. Aby nám obranné mechanizmy pred hammeringom (intenzívne bombardovanie SMTP z jedného počítača) neovplyvňovali testy, tak sme ich pred samotným testovaním vypli:

```
smtpd_client_event_limit_exceptions = static:all
```

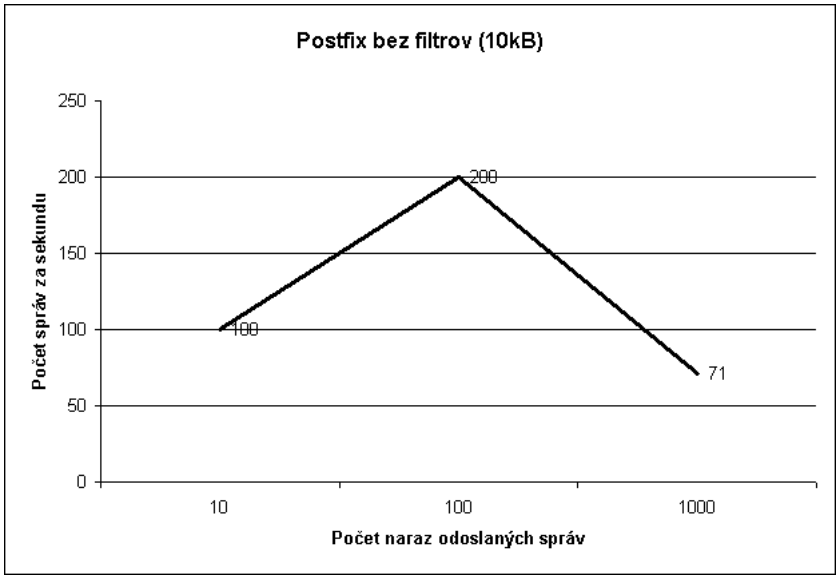
Taktiež boli vypnuté všetky irelevantné filtre.

7.3.1 Postfix bez filtra obsahu

Na začiatok, test výkonnosti samotného Postfixu. Tieto výsledky nám udávajú hornú hranicu, ktorú však s najväčšou pravdepodobnosťou nedosiahneme ani s jedným filtrom obsahu. Na základe tejto referenčnej hodnoty môžeme zistiť ako moc bude náš filter degradovať výkon poštového servera.

Malý email

Počet odoslaných emailov	Počet spojení	Čas (s)
10	10	0,1
100	10	0,5
1000	10	14

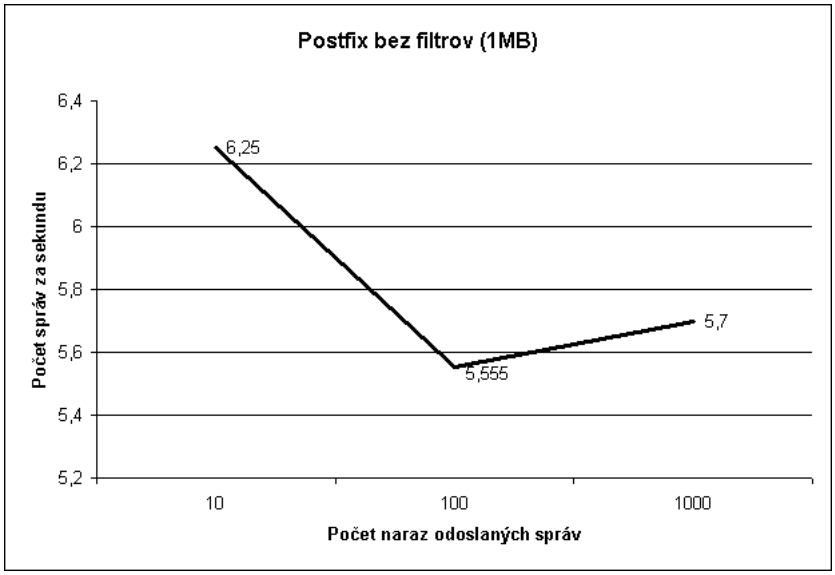


Obr. 7.1: Postfix bez filtrov (10 kB)

Veľký email

Počet odoslaných emailov	Počet spojení	Čas (s)
10	10	1,6
100	10	18
1000	10	174

Analýza Výkon Postfixu pri malých emailoch sa začal prejavovať až pri 100 emailoch. Zaujímavým je degradácia výkonu pri hodnote 1000 emailov, kde je vidieť, že už ani on nestíha. Keďže doba spracovania emailu je priamo závislá na použitom poštovom démonovi ako aj príslušných filtroch, tak môžeme očakávať, že budú filtre vykazovať pri tejto hodnote rapídnu degradáciu výkonu.



Obr. 7.2: Postfix bez filtrov (1 MB)

Pri 1 MB emailoch je vidieť rapídne zníženie výkonu, vzhľadom k počtu naraz prijatých emailov, avšak u hodnoty 1000 sa to začína zlepšovať a to zrejme vďaka lepšej organizácii emailov v rámci front, ktoré Postfix obsahuje. Tento nárast však nie je tak rapídny, aby to nemohla byť chyba pri meraní.

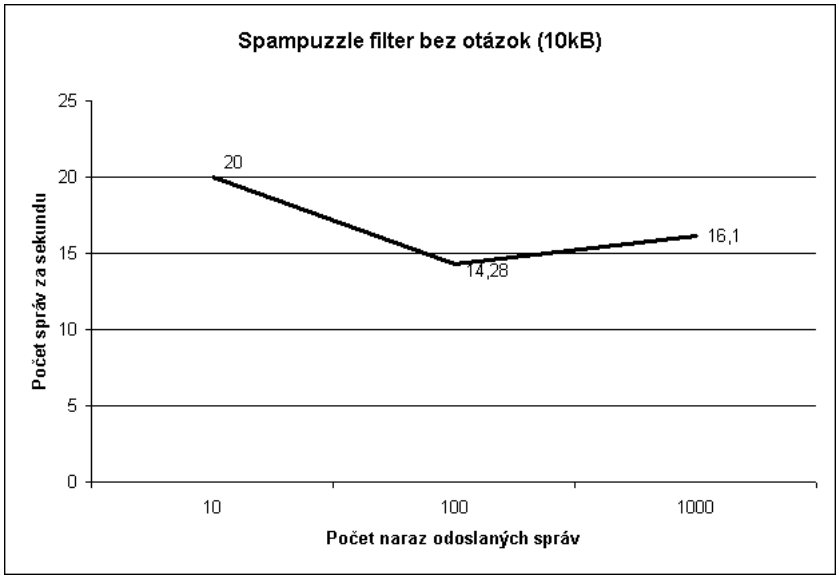
7.3.2 Spampuzzle s deaktivovaným filtrom obsahu

Pri tejto analýze vychádzame z nastavení, ktoré obsahuje Spampuzzle, na začiatku implementácie. Neobsahuje žiadne otázky, a teda jedine čo robí je, preklad pošty z jednej fronty do druhej.

Malý email

Počet odoslaných emailov	Počet spojení	Čas (s)
10	10	0,5
100	10	7
1000	10	62

Veľký email



Obr. 7.3: Spampuzzle s deaktivovaným filtrom obsahu (10 kB)

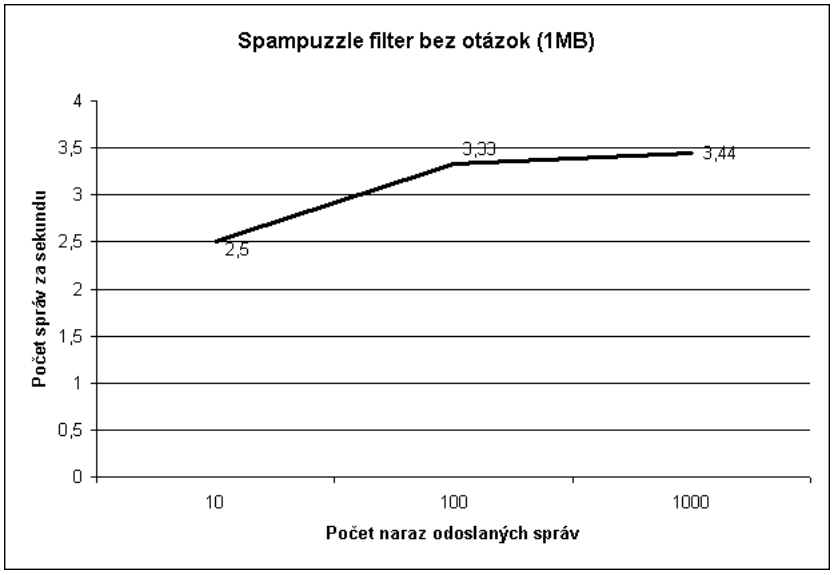
Počet odoslaných emailov	**Počet spojení**	**Čas (s)**
10	10	4
100	10	30
1000	10	290

Analýza Je až zaujímavé sledovať aký veľký dopad má tento preklad medzi smtp frontami, na výkon celého riešenia. Porovnaním grafu Postfixu bez filtru a tohto, tak je vidieť v oboch prípadoch, že akonáhle postfix poľaví na rýchlosti spracovania emailov tak naše riešenie sa naopak zrýchli. Takže by sa z toho dalo vyvodiť, že v našom riešení chýba správca príchodzích emailov, ktorý by pomáhal vybalancovať takéto preťaženia.

7.3.3 Spampuzzle s aktívnym filtrom v režimu zadrž a polož otázku

Spampuzzle s nastavenými otázkami a v režime hold, tzn. že každý príchodzí email, ktorý nie je autentifikovaný našim systémom sa odloží na disk. Vďaka povahe testu, sme prenastavili Spampuzzle na tieto hodnoty

- --max-retry=1000



Obr. 7.4: Spampuzzle s deaktivovaným filtrom obsahu (1 MB)

- --retry-delay=1

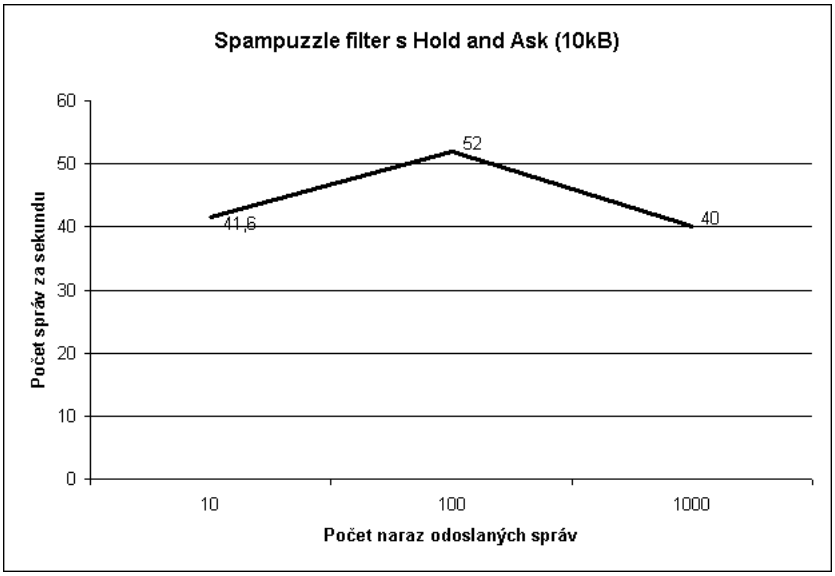
Týmto sme schopný každú sekundu vygenerovať jeden email s otázkou.

Malý email

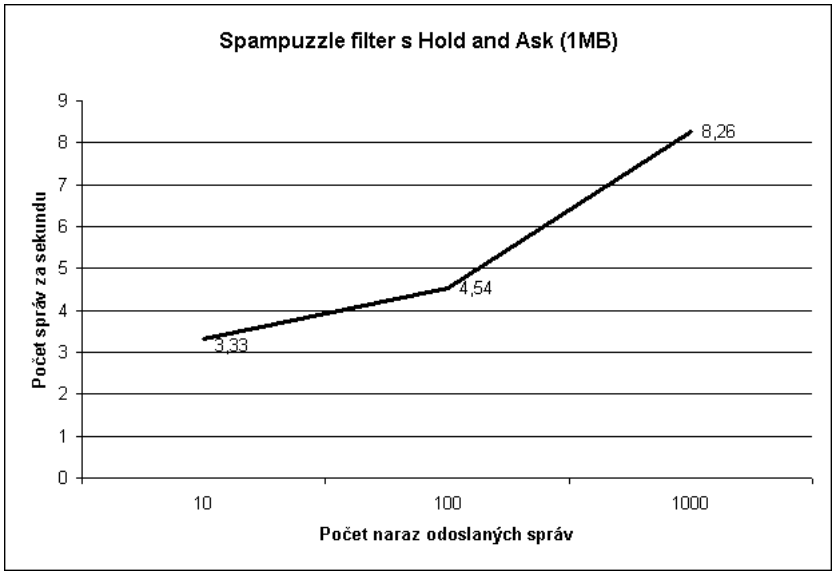
Počet odoslaných emailov	Počet spojení	Čas (s)
10	10	0,3
100	10	2,1
1000	10	25

Veľký email

Počet odoslaných emailov	Počet spojení	Čas (s)
10	10	3
100	10	22
1000	10	221



Obr. 7.5: Spampuzzle s aktívnym filtrom v režimu zadrž a polož otázku (10 kB)



Obr. 7.6: Spampuzzle s aktívnym filtrom v režimu zadrž a polož otázku (1 MB)

Analýza Výsledok je až nadmieru vynikajúci a krivkou podobný tej z postfixu, iba s nižšími hodnotami. U merania s 1MB emailom je zaují-

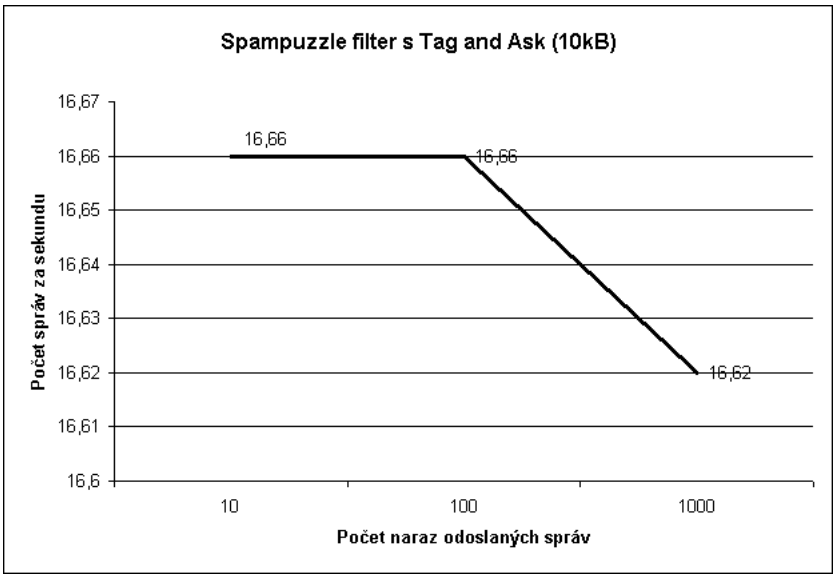
mavé sledovať ako sa krivka vyvíja vzhľadom k počtu emailov odoslaných naraz. Myslím si, že v tomto hrajú úlohu algoritmy filesystému, ktoré pracujú efektívnejšie pri väčšom zaťažení, keďže sa snažia oddialovať zápis na disk.

7.3.4 Spampuzzle s aktívnym filtrom v režimu označ a polož otázku

Spampuzzle je v režime označenia Predmetu reťazcom [UNVERIFIED] a doručením emailu. Samozrejme to sprevádza zadanie otázky odosielajúcej strane.

Malý email

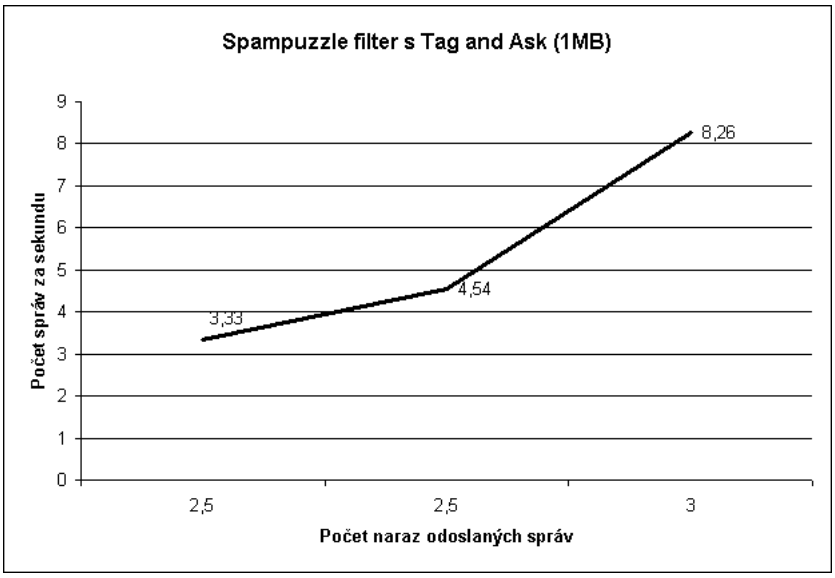
Počet odoslaných emailov	Počet spojení	Čas (s)
10	10	0,6
100	10	6
1000	10	64



Obr. 7.7: Spampuzzle s aktívnym filtrom v režimu označ a polož otázku (10 kB)

Veľký email

Počet odoslaných emailov	Počet spojení	Čas (s)
10	10	4
100	10	40
1000	10	332



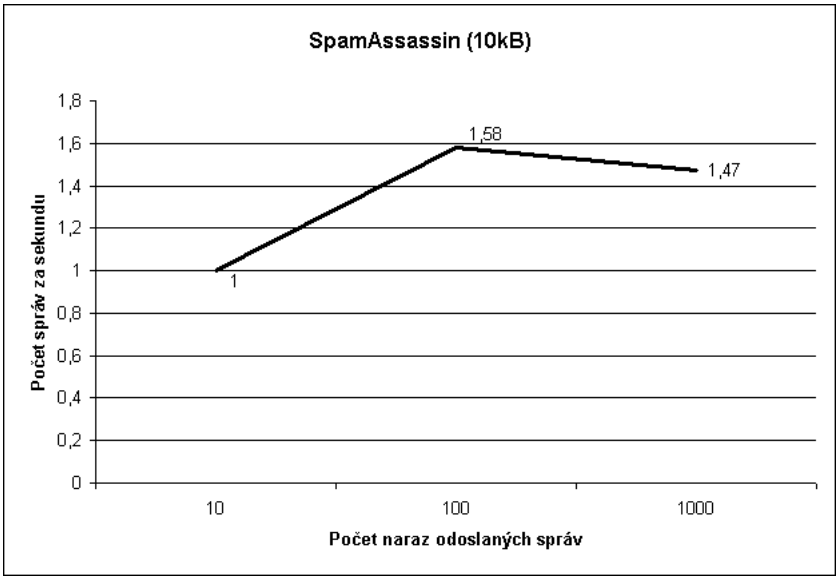
Obr. 7.8: Spampuzzle s aktívnym filtrom v režimu označ a polož otázku (1 MB)

Analýza Táto metóda vykazovala počas celého merania veľmi stabilné výsledky.

7.3.5 Postfix s aktívnym filtrom obsahu SpamAssassin

Malý email

Počet odoslaných emailov	Počet spojení	Čas (s)
10	10	10
100	10	63
1000	10	679



Obr. 7.9: Postfix so SpamAssassinom (10 kB)

Analýza Meranie SpamAssassinu nám oproti prázdnemu postfixu ukazuje jednoznačne dolnú hranicu časovej náročnosti spracovania emailu, ktorá sa v tomto prípade pohybuje medzi jedným až dvoma emailami za sekundu. To je viac ako 100 násobné spomalenie v určitých situáciách.

Kapitola 8

Záver

8.1 Zhodnotenie

Spam spôsobuje firmám a jednotlivcom každoročne obrovské finančné straty, ktoré sa postupom času zväčšujú. V rámci práce sme mohli vidieť viacero spôsobov, ktoré sa pokúšali tento nepriaznivý trend pozastaviť avšak aj za cenu zdržania emailu sa objavujú v poštových schránkach emaily, o ktoré nemáme záujem.

Zmapovaním situácie v tomto sektore sme zistili, že za účinné riešenie môže byť považovaná metóda greylisting, ktorá šetrí zdroje a núti spammera poslať email ešte raz. Avšak zdroje spammera sa postupom času spolu s vývojom počítačovej sféry ako aj s nárastom počtom užívateľov Internetu zväčšujú. Mimo to v niektorých prípadoch môže byť oneskorenie príliš dlhé na to, aby to boli ochotné firmy, prípadne jednotlivci, zaplatiť.

Preto sa oči administrátorov upierajú na nové technológie akými sú DKIM a SPF. Tie však vyžadujú pre svoju účinnosť masovejšieho zapojenia ľudí, ktorý ju budú podporovať. Bez toho nemajú tieto technológie až také uplatnenie.

Spampuzzle sa pokúša priniesť niečo čo nemusí byť nutne spojované s masovým rozširovaním a taktiež nie je potreba implementácie na druhej strane (u príjemcu). Vďaka jednoduchým otázkam, dokážeme našim turingovým testom, ako v prípade testu CAPTCHA, rozlíšiť človeka od robota a tým vpustiť iba emaily, ktoré sú odosielané aktívnym človekom. Ten sa preukáže tým, že zodpovie na túto otázku. Táto technológia môže presvedčiť ľudí, ktorí už stratili dôveru k emailu (kvôli miere spamu, ktorý ich obťažuje), aby začali znovu využívať email ako predtým, keď sa nemusela zaoberať spamom. Oproti technológiám ako Bayesovské fil-

tre je jej celý princíp v podstate jednoduchý a vďaka tomu a možnosti konfigurovateľnosti je systém vhodný pre každého. Kúzlo celého systému je pritom založené na dotazu a odpovedi. tak jednoduchom fakte, akým sú zvolené otázky podľa požiadaviek užívateľov. Tie pritom pomáhajú vyčleniť nielen spamy, ale aj prípadných ľudí, ktorí nepatria do nášho blízkeho okolia, tzn. nepoznajú naše prostredie a návyky, na ktoré sa v otázke môžeme odkazovať. Efektivitu môžeme zvyšovať obmieňaním otázok podľa potreby. Tým dosiahneme takmer bez údržbový systém, ktorý znemožní prepustenie spamu do našej poštovej schránky.

Riešenie sa snaží položiť latku dostatočne vysoko, aby odradilo spammerov. Pomalým zvyšovaním latky sa dokážu prispôsobovať dostatočne rýchlo a tým neustále prekonávať nové technológie. Následkom tohto zvýšeného nároku na konto odosielateľa, môže vzniknúť nevôľa niektorých užívateľov prijímať takúto formu autentifikácie. Každopádne si myslím, že aj napriek tomu je táto cena stále prijateľná pri porovnaní, čo za ňu dostaneme. A to metódu, ktorá nevykazuje v aktuálnych podmienkach slabiny, nakoľko stojí na princípe sémantickej analýzy, ktorá je pre počítače príliš zložitá a jej riešenie je v nedohľadne.

Na základe vykonaných testov môžeme ohodnotiť výkon riešenia ako nadmieru postačujúci pre tradičný segment, nakoľko v porovnaní so spamassassinom je aj pri náročných nastaveniach, niekoľko násobne výkonnejšie.

V budúcnosti by sa mohla implementácia naďalej zdokonaľovať a to hlavne v oblasti užívateľskej interakcie, kde by sa vďaka úplnej implementácie filozofie Client Puzzle dokázala odbúrať nutná dávka užívateľského vstupu pri overovacom procese. Užívateľ by dostal email a jeho agent by si sám vypočítal potrebné matematické úlohy a odoslal ich späť serveru. Celý proces by bol týmto pre užívateľa transparentný. Otázkou však ostáva ako ťažké úlohy je treba zvoliť a či takéto počítanie nakoniec nezdraží poštu vo všeobecnosti, keďže spotreba počítačových komponent postupne prekonáva všetky hranice, a hardware začína byť lacnejší ako energia, ktorú spotrebuje.

Literatúra

[1] *Mail (MX) Server Survey* [online]. 2006 , August 1st, 2006 [cit. 2008-08-02]. Dostupný z WWW: <http://www.securityspace.com/s_survey/data/man.200607/mxsurvey.html>

[2] SIMPSON, Ken, BEKMAN, Stas. *Fingerprinting the World's Mail Servers* [online]. 2007, 01/05/2007 [cit. 2008-08-02]. Dostupný z WWW: <http://www.oreillynet.com/pub/a/sysadmin/2007/01/05/fingerprinting-mail-servers.html>

[3] MANION, Art, HERNAN, Shawn V. , LANZA, Jeffery P. *CERT Advisory CA-2003-12 Buffer Overflow in Sendmail* [online]. Carnegie Mellon University : CERT, 2003 , May 29, 2003 [cit. 2008-08-01]. Dostupný z WWW: <http://www.cert.org/advisories/CA-2003-12.html>.

[4] MANION, Art. *CERT Advisory CA-2003-25 Buffer Overflow in Sendmail* [online]. Carnegie Mellon University : CERT, 2003 , September 29, 2003 [cit. 2008-08-01]. Dostupný z WWW: <http://www.cert.org/advisories/CA-2003-25.html>.

[5] *IC3 2005 Internet Fraud Crime Report : January 1, 2005-December 31, 2005* [online]. Internet Crime Complaint Center, 2006 [cit. 2008-08-01]. Dostupný z WWW: <http://www.ic3.gov/media/annualreport/2005_IC3Report.pdf>.

[6] *Spam To Be Canned By 2006 : Microsoft Chairman Announces Plans To End Junk E-Mail.* CBS News [online]. 2004 [cit. 2008-08-01]. Dostupný z WWW: <http://www.cbsnews.com/stories/2004/01/24/tech/main595595.shtml>.

[7] *CAN-SPAM Act of 2003* [online]. USA : 2003 , 1.3.2007 [cit. 2008-08-01]. Text v angličtine. Dostupný z WWW: <http://uscode.house.gov/download/pls/15C103.txt>.

[8] *Europoslanci chcú zdaniť e-maily a SMS.* Pravda [online]. 2006 [cit. 2008-08-01]. Dostupný z WWW: <http://tvojepeniaze.pravda.sk/sk_pspravy.asp?c=A060526_171926_sk_pspravy_p01>.

[9] HIGGINS, Kelly Jackson. *New Massive Botnet Twice the Size of Storm : 400,000-strong 'Kraken' botnet has infiltrated 50 Fortune 500 companies – and now usurps Storm as world's biggest botnet* [online]. 2008 , APRIL 7, 2008 [cit. 2008-08-01]. Dostupný z WWW: <http://www.darkreading.com/document.asp?doc_id=150292&WT.svl=news1_1>.

[10] *Fortune 500* [online]. 2008. USA : Fortune, 2008 , May 5, 2008 [cit. 2008-08-01]. Dostupný z WWW: <http://money.cnn.com/magazines/fortune/fortune500/2008/full_list/>.

[11] *Kaspersky Lab detects the first mass mailing of MP3 spam* [online]. Kaspersky Lab, 2007 , 18 Oct 2007 [cit. 2008-08-01]. Dostupný z WWW: <http://www.kaspersky.com/news?id=207575575>.

[12] KEIZER, Gregg. *Dutch Botnet Suspects Ran 1.5 Million Machines* [online]. TechWeb Technology News, 2005 , October 21, 2005 [cit. 2008-08-01]. Dostupný z WWW: <http://www.techweb.com/wire/security/172303160>.

[13] KEIZER, Gregg. *RSA - Top botnets control 1M hijacked computers : They can dump more than 100B spam messages on users daily* [online]. Computerworld, 2008 , 10/04/2008 [cit. 2008-08-01]. Dostupný z WWW: <http://www.computerworld.com.au/index.php/id;1183357273>

[14] *MS Exchange 2003 SP2 bug related to greylisting* [online]. 2007, May 2007 [cit. 2008-08-04]. Dostupný z WWW: <http://www.greylisting.org/forums/showthread.php?tid=18>

[15] JUELS, Ari, BRAINARD, John. *Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks* [online]. 2007, [cit. 2008-08-05]. Dostupný z WWW: <http://www.rsa.com/rsalabs/staff/bios/ajuels/publications/client-puzzles/clientpuzzles.ps>.

[16] ALLMAN, E., CALLAS, J., DELANY, M., LIBBEY, M., FENTON, J., THOMAS, M. *DomainKeys Identified Mail (DKIM) Signatures* [online]. 2007, May 2007 [cit 2008-08-04]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc4871.txt>.

[17] DELANY, M. *Domain-Based Email Authentication Using Public Keys Advertised in the DNS (DomainKeys)* [online]. 2007, May 2007 [cit 2008-08-04]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc4870.txt>.

[18] POSTEL, Jonathan B. *SIMPLE MAIL TRANSFER PROTOCOL* [online]. 1982, August 1982 [cit 2008-08-04]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc0821.txt>.

[19] *The Postfix Home Page* [online]. 2008 [cit. 2008-08-02]. Dostupný z WWW: <http://www.postfix.org/>

[20] *Where in the World does all the spam come from??* [online]. Spam Blocker Software Co., 2007 [cit. 2008-08-02]. Dostupný z WWW: <http://spameater.com/countries-where-spam-comes-from.php?r=0>.

[21] *XE-Filter: Spam Origin Study : Filtering Spam eMail by its "Country of Origin"* [online]. Computer Mail Services, Inc., 2006 , November 2006 [cit. 2008-08-02]. Dostupný z WWW: <http://www.cmsconnect.com/xefpro/Lib/XEFSpamPercent.htm>.

[22] IVERSON, Al. *Spamhaus ZEN: The DNSBL Resource Review* . DNSBL Resource, 2007 , October 12, 2007 [cit. 2008-08-02]. Dostupný z WWW: <http://www.dnsbl.com/2007/10/spamhaus-zen.html>.

[23] CROCKER, Dave. *DKIM Frequently Asked Questions* [online]. 2007 , 16-Oct-2007 [cit. 2008-08-02]. Dostupný z WWW: <http://www.dkim.org/info/dkim-faq.html#related>.

[24] EVETT, Don. *Spam Statistics 2006* [online]. 2006 [cit. 2008-08-01]. Dostupný z WWW: <http://spam-filter-review.toptenreviews.com/spam-statistics.html>.

[25] KULIKOVA, Tatyana. *Spam Report: June 2008* [online]. 2008, Aug 06 2008 [cit. 2008-08-07]. Dostupný z WWW: <http://www.viruslist.com/en/analysis?pubid=204792015>.

[26] NBBN. *Session Hijacking Vulnerability* [online]. 2008, 13 Mar 2008 [cit. 2008-08-03]. Dostupný z WWW: <http://packetstormsecurity.org/0803-exploits/phpbb2023-hijack.txt>.

[27] HOPE, Jessica. *XSS in admin logs* [online]. 2008, July 06th 2008 [cit. 2008-08-03]. Dostupný z WWW: <http://packetstormsecurity.org/0807-exploits/vbulletin-adminxss.txt>