Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE

Jan Nepožitek

Antispamming na principu výběru vyžádaných zpráv

Katedra softwarového inženýrství

Vedoucí diplomové práce: RNDr. Ing. Jiří Peterka
Studijní program: Informatika

Věnování

Tuto diplomovou práci věnuji svému milovanému bratrovi Peťovi, který mě navždy opustil těsně před jejím dokončením.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 9.8.2007 Jan Nepožitek

Obsah

Abstrakt

Název práce: Antispamming na principu výběru vyžádaných zpráv
Autor: Jan Nepožitek
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí diplomové práce: RNDr. Ing. Jiří Peterka
e-mail vedoucího: peterka@mff.cuni.cz

Abstrakt: Cílem této diplomové práce je prozkoumat možnosti řešení založené na opačném principu, než je většina dnešních metod boje proti spamu - na výběru vyžádaných zpráv z celkového toku přijímaných zpráv. Na začátku práce jsou popsány varianty uvažovaných řešení a výhody i nevýhody stávajících antispamových metod. Následuje podrobnější popis a analýza metod týkajících se výběru vyžádaných zpráv a nakonec i návrh koncepce vlastního řešení.
Součástí práce je též implementace navrženého systému, včetně popisu použitých metod a programů. Na závěr je uvedena analýza výhod a nevýhod uvedeného postupu, možnosti budoucího rozšíření a rovněž návod k instalaci.
Klíčová slova: Spam, Antispamming, Internet, Email

Title: Fighting spam through selecting wanted messages
Author: Jan Nepožitek
Department: Department of Software Engineering
Supervisor: RNDr. Ing. Jiří Peterka
Supervisor's e-mail address: peterka@mff.cuni.cz

Abstract: The goal of this thesis is to look into the possibilities of solution based on the opposite principle than most of today's antispam solutions - selecting wanted messages from the total flow of all incoming messages. At the beginning of the thesis are described variants of calculated solutions and advantages and disadvantages of current antispam methods. Then follows a more detailed description and analysis of methods relating to selecting wanted messages and finally a design of own solution.
The part of the thesis is also an implementation of the suggested system including a description of used methods and programs. In conclusion is mentioned an analysis of advantages and disadvantages of the suggested procedure, the possibilities of future improvements and also install instructions.
Keywords: Spam, Antispamming, Internet, Email

1 Úvod

V této kapitole jsou popsány cíle diplomové práce, její zadání, možnosti řešení, struktura textu a obsah jednotlivých kapitol.

1.1 Zadání diplomové práce

Zadáním této diplomové práce bylo prozkoumat možnosti řešení založené na opačném principu, než je většina dnešních metod boje proti spamu - na výběru chtěných (vyžádaných) zpráv z celkového toku přijímaných zpráv. Dále se měla navrhnout koncepce takového řešení a její základ implementovat.

1.2 Analýza řešení

Prvním cílem této práce bylo stanovit, jak lze vlastně chápat zadání. Obecně výběr vyžádaných zpráv z celkového balíku přijatých zpráv je v podstatě to samé, jako výběr nevyžádaných zpráv.V jednom případě se za vyžádanou zprávu označuje to co se vybere a ve druhém to co zůstane po odstranění nevyžádaných zpráv. V případě navržení systému pouze na bázi této myšlenky by se jednalo o inverzní postup, který by nepřinesl mnoho nového.

Dále bylo potřeba prozkoumat, na jaké úrovni systém nejlépe navrhnout. V úvahu připadalo řešení buď na straně poštovního serveru, nebo na úrovni uživatele. V první variantě se systém aplikuje na všechnu příchozí poštu a to buď během procesu přijímání, nebo následně po jeho ukončení. Ve druhé variantě probíhá zpracování na úrovni jednotlivých uživatelů a to po výběru zpráv ze schránky na serveru pomocí poštovního klienta.

Neméně důležitou částí byla též analýza stávajících řešení a z ní vyplývající úvaha, jestli je lepší systém navrhnout jako monolitické řešení, nebo ho sestavit z již dostupných částí.

1.3 Uvažované možnosti řešení

První věc, kterou bylo potřeba rozhodnout, bylo umístění předpokládaného systému, tedy na jaké úrovni bude běžet.

V případě uvedené varianty, kdy program běží na úrovni uživatele, by se zřejmě jednalo o pouhé filtrování neboli přerozdělení pošty do složek. Z poštovní schránky na serveru je totiž nejprve stáhnuta všechna doručená pošta a až s tou se posléze pracuje. Pokud se bude jednat o vybírání vyžádaných zpráv z této doručené pošty, tak v tomto případě to je úplně to samé jako eliminace zpráv nevyžádaných. Jediné možné variace systémů na této úrovni jsou metody pro ohodnocení pošty a následné akce s takto ohodnocenými emaily.

Z tohoto důvodu se všechny následně uvažované možnosti týkaly zpracování zpráv na straně poštovního serveru.

Jako první myšlenka byl systém postavený na metodě Challenge-response. V principu se jedná o vytyčení „cesty“[1] od odesílatele k příjemci. Popis metody bude uveden v další kapitole, základní idea je ovšem taková, že email z neznámé adresy není přijat ihned, ale až po splnění určité podmínky, zatímco email ze známé adresy (tj. adresa, ze které už byl doručen nějaký email) je přijat ihned. Tedy se vytváří určitá databáze adres, ze kterých jsou emaily považovány za vyžádané.

Jednalo by se o poštovní server, který by na veškerou příchozí poštu aplikoval uvedenou metodu a k uživateli by pouze propustil zprávy, jenž by systémem úspěšně prošly. Dále by měl uživatel možnost se kdykoliv podívat do seznamu pozdržených emailů[2], jestli zde náhodou není zpráva, která neprošla systémem, ale pro uživatele představuje legitimní email. Například při objednání zboží po internetu je většinou vygenerován potvrzující email, na který musí uživatel odpovědět. Jelikož je ale tento email vygenerován automatickým systémem, který neumí z principu splnit podmínku metody Challenge-response, tak tento email uživatel přímo do schránky nedostane, byť je pro něj vyžádaný.

[1] lze chápat jako vytvoření důvěryhodné dvojice adres (email odesílatele, email příjemce), kde předpokládáme, že všechny emaily z adresy odesílatele budou u příjemce považovány za vyžádané.
[2] doručené emaily, u kterých se čeká na splnění podmínky metody Challenge-response.

V tomto případě se ale nedá mluvit o zásadní nevýhodě, jelikož uživatel automaticky generovanou zprávu očekává a tedy ví, že ji nemá hledat ve své schránce, ale je umístěna v seznamu nepotvrzených zpráv. Při bližším zkoumání této metody a možností jejího vylepšení bylo ovšem postupem času nalézáno čím dál tím více nevýhod samotného principu, někdy i zásadních a proto byl systém založený na této myšlence nakonec zavrhnut. Zmiňované nevýhody a samotný popis metody je uveden v kapitole 2.4.

Jako další následně uvažovaná varianta byl samostatný poštovní server, který by pouze plnil úlohu směrování a všechnu příchozí poštu by rozděloval mezi jiné dva poštovní servery na základě vhodně zvolené metody pro ohodnocení emailů.

Po ohodnocení příchozího emailu by byla zpráva v případě kladného ohodnocení předána poštovnímu serveru, kde by měl uživatel prioritní schránku a v opačném případě by putovala na server, kde by měl uživatel spamovou schránku. Došlo by tedy k oddělení nevyžádané pošty od vyžádané ještě před doručením do samotné schránky uživatele a tedy by se účinně reguloval tok pošty do prioritní schránky s tím, že by si občas uživatel zkontroloval obsah spamové schránky, jestli se zde náhodou nevyskytuje legitimní zpráva.

Jako vhodná metoda pro ohodnocení příchozích zpráv byla nakonec zvolena Bayesova analýza (kapitola 2.3), která vykazuje velice dobré výsledky a též výborný poměr přijaté spamy/nedoručené legitimní zprávy.

Při podrobnější analýze bylo nakonec zamítnuto i toto řešení, jelikož principielně skoro totéž lze udělat přesměrováním na jiné dvě adresy v rámci téhož poštovního serveru. Tím by navíc odpadla nutnost instalování dalšího poštovního serveru a s tím spojená nutná režie pro jeho údržbu. Navíc by se v případě instalace tohoto předsunutého směrovacího poštovního serveru musely též změnit MX záznamy dané domény pro směrování příchozí pošty na tento server.

Další uvažovanou možností bylo vytyčení již zmiňované „cesty“ od odesílatele k příjemci pouze na žádost odesílatele o komunikaci, kdy ještě před započetím vlastní komunikace musí příjemce potvrdit, že s ní souhlasí. Toto potvrzení lze například prostřednictvím webové adresy, na které odesílatel vyplní svoji emailovou adresu, na níž mu po schválení přijde emailová adresa příjemce (případně potvrzující email z adresy příjemce), nebo zasláním emailu v předem dohodnutém formátu na již známou adresu, kde se email následně zpracuje.

Již z principu se ovšem toto řešení vůbec nehodí jako samostatně fungující systém, například pro příliš komplikované navazování prvotní komunikace a s tím spojenou režii ze strany jak odesílatele tak i příjemce.

Jako poslední možnost, která se nakonec ukázala jako nejvíce vhodná, byl systém postavený na principu proxy, která bude řídit tok pošty od odesílajícího poštovního serveru k přijímacímu poštovnímu serveru a to na základě upravených existujících metod pro ohodnocení emailů. Toto řešení je popsané v samostatné kapitole 3 týkající se návrhu vlastního řešení.

1.4 Struktura diplomové práce

1.4.1 Terminologie

Pro nevyžádané zprávy je v textu používán též anglický termín spam, pro vyžádané zprávy anglický termín ham, případně český výraz nespam.

Názvy uvedených metod boje proti nevyžádané poště jsou v dalším textu ponechány v angličtině:

challenge-response – výzva-odpověď
whitelist – bílý seznam
blacklist – černý seznam
greylisting – šedý seznam

1.4.2 Obsah jednotlivých kapitol

Ve **druhé kapitole** je stručný úvod do spamové problematiky, přehled nejpoužívanějších metod boje proti spamu, stručné informace o tom na jakém základě fungují a případné výhody a nevýhody. Dále jsou v samostatných podkapitolách podrobněji rozebrány další tři metody, jenž se přímo či nepřímo týkají samostatného návrhu vlastního řešení.

Ve **třetí kapitole** je popsáno zvolené řešení, dále je uveden základní návrh programu a přehled o jeho funkcích.

Čtvrtá kapitola popisuje vlastní implementaci programu, dále se zabývá programátorskou částí práce a podává přehled o celkové koncepci řešení.

Pátá kapitola se věnuje zhodnocení práce, dosažení stanovených cílů, srovnání s ostatními metodami, možnostmi vylepšení a budoucího vývoje.

1.4.3 Přílohy

V **příloze A** je popis použitých programů třetích stran.

V **příloze B** je uveden obsah přiloženého cd a popis zdrojových kódů.

V **příloze C** je návod k instalaci programu a všech potřebných součástí.

2 Techniky boje proti spamu

V této kapitole je přehled nejpoužívanějších metod boje proti spamu, stručné informace o tom na jakém základě fungují a případné výhody a nevýhody. Dále jsou v samostatných podkapitolách podrobněji rozebrány další tři metody, jenž se přímo či nepřímo týkají samostatného návrhu vlastního řešení.

2.1 Úvod

Spam je nežádoucí masově šířené reklamní sdělení šířené zejména v prostředí internetu. V prvních počátcích se tento název používal výhradně pro reklamní emaily všeho druhu a dnes je spíše chápán jako označení všech forem obtěžující komunikace, tedy nejen samotného emailu, ale i nežádoucích příspěvků na diskusních fórech, zneužití programů pro výměnu krátkých zpráv, atd.

Historie spamu se začala psát již od roku 1978, kdy byl zaslán zřejmě první hromadný email s reklamním sdělením v síti ARPANET. V dnešní době tvoří spam podle některých zdrojů až 90% nevyžádané pošty ([1]), navzdory neustálému boji, jenž je proti němu veden. Jedna z největších výzev problematiky spamu je vývoj co nejdokonalejšího systému pro eliminaci spamu, kde je nejen důležité, kolik spamu daný systém dokáže rozpoznat, ale zejména i poměr mezi zachyceným spamem a nesprávně označenými legitimními emaily. Pro uživatele je mnohdy lepší dostat 100 nevyžádaných emailů místo nedoručení jednoho legitimního emailu.

Ve zbytku této kapitoly jsou stručně popsány nejznámější metody a návrhy boje proti nevyžádané poště.

2.2 Přehled nejznámějších antispamových metod

Analýza obsahu pomocí klíčových slov

Jeden z nejstarších způsobů eliminace spamu zahrnuje studium předmětu a těla zprávy. Z původních jednoduchých filtrů detekující pouze přesně zvolená slova se vyvinuly mnohem komplexnější filtry, které si dokáží poradit i s různým maskováním, jako je

například mnoho interpunkce na jednom místě, náhrada písmena písmenem jemu vizuálně podobným, nebo i vkládané značky jazyka HTML. Nevýhodou těchto metod bylo už od začátku velké množství nesprávně označeného spamu a postupem času čím dál menší účinnost.

Filtrování na bázi pravidel

Oproti jednoduché detekci slov indikující možný spam navíc vytváří sadu heuristických pravidel sledujících vazby mezi jednotlivými slovy, co znamenají a jaký je mezi nimi vztah. Rozšiřuje se tedy rozhodovací proces, kterým každá zpráva prochází.

Whitelist

Vytváří se seznam[3] známých odesílatelů, jejichž adresy jsou považované za důvěryhodné. I když z těchto adres bude odeslán email, který by mohl být označen jako spam, tak má právo projít bez filtrování. Tato metoda je stále používána, nehodí se ovšem jako samostatné řešení, ale pouze jako doplněk k jiným metodám.

Blacklist

Vytváří se seznam IP adres, ze kterých byl odeslán nějaký spam. V dnešní době je tato technika použitelná snad jenom pro open relay[4] servery, jelikož naprostá většina odeslaného spamu je pokaždé odeslána z různých adres (například přes napadené počítače). Také je zde velice časté nesprávné zařazení IP adresy poštovního serveru do blacklistu, kdy jsou posléze blokovány všechny emaily z této domény.

Záměrné zpomalení SMTP[5] komunikace

Efektivita spamů je velice malá a proto je musí spameři posílat ve velkém množství. V případě, že by odeslání jednoho emailu trvalo delší dobu, tak by celková efektivita nesmírně poklesla a posílání spamů by přestalo být výhodné. Na druhou stranu by se

[3] lze buď na úrovni poštovního serveru, kdy se zaznamenají IP adresy odesílajících poštovních serverů, nebo na úrovni uživatele, kdy je seznam tvořen emailovými adresami jednotlivých uživatelů.
[4] nastavení poštovního serveru, jenž umožňuje každému na Internetu odesílat přes něj emaily.
[5] Simple Mail Transfer Protocol - internetový protokol určený pro přenos zpráv elektronické pošty mezi jednotlivými stanicemi.

nesměla omezit rychlost odesílání emailů u poštovních serverů velkých ISP[6], jelikož zde dochází k velkému objemu přenášené pošty a rychlost přenosu je tedy zásadní a nezbytně nutná.

V každém případě tohle řešení vyžaduje nový protokol pro posílání emailů a nelze obecně aplikovat na základě stávajícího protokolu SMTP.

Signatury

Filtrování založené na signaturách funguje na principu porovnání příchozího emailu s databází známých spamů. Signatura emailu vznikne například tak, že každý znak ve zprávě ohodnotíme číslem a tyto čísla nakonec spojíme v jedno číslo. Dva identické emaily mají stejnou signaturu, tedy při porovnání oproti databázi signatur lze zjistit, zda databáze obsahuje testovaný email či nikoliv. Velkou nevýhodou signatur je možnost vložení náhodného textu do každého spamu, čímž vznikne jiná signatura a tím pádem se tato metoda stane neúčinnou.

Označkovaná adresa

Metoda označkované adresy spočívá ve speciálním formátu adresy, do které je zakódována dodatečná informace (značka), podle které se určí legitimnost emailu. Tato značka v sobě nese například informaci o čase, po kterou má emailová adresa platnost, odesílateli, od kterého lze přijmout email, atd.

Tento formát adresy se hodí zejména pro různé konference, kde potřebujeme emailovou adresu pouze na určitou dobu a chceme emaily dostávat pouze od určité skupiny uživatelů. V případě, kdy na ni začne chodit spam, ji lze jednoduše zrušit a nahradit případně jinou.

Metodu ovšem nelze použít univerzálně, například z důvodu špatně zapamatovatelného formátu adres, či omezené platnosti.

[6] Internet Service Provider – poskytovatel internetového připojení

Zpoplatnění emailu

Koncept, kdy bude zpoplatněna veškerá emailová komunikace nějakou sazbou, která je zanedbatelná pro normálního uživatele odesílajícího pár emailů denně, ale přinese už značné náklady na straně spammera odesílajícího miliony zpráv.

Velkým problémem je ovšem chybějící infrastruktura pro výběr těchto poplatků (opět nutnost nového protokolu) a velká hrozba zneužití, kdy z napadeného počítače nic netušícího uživatele odejde obrovské množství spamu a finanční „postih" odnese místo spammera třetí strana.

Stížnosti poskytovateli ISP, zákony proti spamu, …

Mezi další formy boje patří nejenom softwarová řešení, ale například i stížnosti ISP, z jehož domény chodí spamy (případně kam se odkazují spamy), nebo z právnického hlediska antispamové zákony, kde spammerům hrozí nejen finanční postih. Tyto a další možnosti jsou mimo zaměření této práce, takže zde nebudou popsány.

2.3 Bayesovské filtry

Bayesovské[7] filtry dnes patří mezi nejúčinnější řešení. Metoda se začala rozvíjet na základě článku „A plan for spam"od Paula Grahama [7], který byl publikován v roce 2002. Princip spočívá v analýze jednotlivých tokenů[8], jenž emaily obsahují. Bayesovský filtr začíná se dvěmi kolekcemi emailů a to s kolekcí spamů a kolekcí nespamů. Pro všechny tokeny v nich obsažené se spočítá hodnota pravděpodobnosti, která bere v úvahu četnost výskytu slova v nevyžádané poště (kolekce spamů) vzhledem k legitimní poště (kolekce nespamů).

Z každého příchozího emailu se vybere 15 až 20 (může se lišit dle konkrétní implementace) tokenů s nejvyšší odchylkou od neutrální pravděpodobnosti[9] a celkové ohodnocení pro daný email se vypočte následovně: pravděpodobnosti vybraných slov se vynásobí a z této hodnoty se vypočítá geometrický průměr (odmocní se tolikátou odmocninou, kolik se

[7] Bayesova analýza je statistická teorie pravděpodobnosti objevená v 18.století Thomasem Bayesem. Určuje pravděpodobnost budoucích jevů na základě již shromážděných vzorků.
[8] Většinou seskupení znaků,čísel,čárek a apostrofů. Záleží ovšem na konkrétní implementaci každého filtru.
[9] Hodnota blížící se číslu 0,5 a značící, že dané slovo se vyskytuje přibližně stejně často jak ve spamu tak i v legitimním emailu a tedy nemá vypovídající hodnotu pro klasifikaci emailu.

násobilo členů). Čím více se výsledné číslo blíží nule, tím je pravděpodobnější, že testovaný email je vyžádaná zpráva a naopak, čím více se výsledek blíží jedničce, tím pravděpodobněji půjde o spam.

Na rozdíl od jiných metod se rozpoznávají nejenom slova identifikující spam, ale i slova, jenž označují validní zprávu. Tedy i případný výskyt nějakého slova v nespamu, jenž se jinak vyskytuje zejména ve spamu, neovlivní celkový výsledek, jelikož tento výsledek ovlivňují i „dobrá" slova. Tyto dobrá slova jsou různá pro každého uživatele (neboli místo, kde je instalován antispamový filtr používající Bayesův klasifikátor) a tedy je pro spammera velice těžké, ba přímo někdy nemožné je znát či uhodnout, aby pomocí jejich vložení do spamu získal celkově kladné ohodnocení. Dokonce i v případě, že by se mu to povedlo, tak uvedená slova, jenž se začnou stále mnohem více objevovat ve spamu, ztratí svůj význam a naopak se může stát, že brzy začnou být více charakteristické pro spam.

Důležitá je též analýza hlavičky emailu, nejenom samotného textu obsahu zprávy. Pravděpodobnosti stejných slov se dosti často liší pouze tím, kde jsou umístěny. Výskyt nějakého slova v těle zprávy může být stejně tak častý ve spamu i nespamu, ale pokud je toto slovo umístěno například v předmětu zprávy, tak může být pravděpodobnost výskytu ve spamu daleko větší.

Velkou výhodou Bayesovkých filtrů je tedy adaptace na nové typy spamu formou „přetrénování" databáze pomocí aktualizovaných kolekcí spamů[10] a nespamů a dále minimální množství špatně označených emailů. Čím větší jsou tyto kolekce, tím přesnější budou výsledky.

Menší nevýhodou je nutnost vytvoření této počáteční databáze dostatečného počtu emailů potřebného k vytrénování, což trvá nějaký čas a během této doby pracuje systém méně efektivně. Lze také použít unifikované kolekce, jenž se většinou dodávají s antispamovým systémem a není tedy nutné nejprve nashromáždit dostatečný počet emailů pro vytrénování. Tyto data ale mají 2 zásadní vady. Za prvé jsou volně přístupná, čehož využívají spammeři pro napsání spamů, jenž jimi projdou a za druhé se obsah obou kolekcí někdy i dost liší od běžné emailové komunikace na instalovaném místě, hlavně co se týká kolekce nespamů. Tedy se systém stane celkově méně efektivním a hrozí větší riziko nesprávného označení legitimní zprávy.

[10] Jedna z elegantních možností, jak nashromáždit velké množství spamu bez obav o nesprávné zařazení legitimní zprávy do spamové kolekce je vytvoření fiktivních emailových účtů s lehce uhodnutelnými názvy. Veškeré příchozí emaily na tyto adresy lze rovnou považovat za spam.

2.4 Challenge-response systémy

Challenge-response systémy jsou založeny na následujícím principu. Příjemce udržuje databázi známých odesílatelů a v případě, že přijde mail od odesílatele, který není uveden v této databázi, tak se mail pozdrží a odesílateli se pošle tzv. výzva, což je vlastně požadavek na autorizaci a má většinou za úkol prokázat, že odesílatelem byla lidská bytost a ne nějaký spamový nástroj. Existuje spoustu forem těchto autorizací, mezi než patří například opsání nějakého textového řetězce z obrázku, nebo navštívení určité webové stránky a nějaká operace na ní.

Pokud odesílatel zareaguje správně na tuto výzvu, tak je odesílatel přidán do databáze mezi známé odesílatele, mail na který se generovala výzva je doručen a každý další mail z této adresy je již považován za důvěryhodný a je automaticky propuštěn.

Na první pohled celkem úspěšná metoda má však spoustu úskalí a nevýhod:

- možnost zneužití whitelistu

  Pokud se jednou adresa přidá do whitelistu, tak od tohoto okamžiku se považuje za důvěryhodnou a všechny emaily z ní poslané se dostanou ihned ke adresátovi. Co se však stane, pokud se autor spamu dostane k tomuto seznamu, případně vysleduje, které adresy by zde mohly být uvedeny?

- lidský faktor

  Existuje spousta lidí, jenž nebude chtít reagovat na výzvu, nepochopí její smysl, případně jim přijde obtěžující či podezřelá.

- napadnutelnost emailového účtu pomocí výzev

  V případě, že autor spamu pošle tisíce mailů, kde jako odesílatel bude uveden podvržený adresát, tak se tahle adresa následně zaplaví neakceptovatelným počtem výzev.

- spolupráce s jinými antispamovými nástroji

Představme si situaci, kdy jak odesílatel, tak i příjemce používají nějaký antispamový nástroj. Buď A odesílatel a B příjemce. A pošle B email, ale protože B používá C-R systém, tak A je ihned odeslána výzva. Zde ovšem nastává otázka, jak si s touto výzvou poradí antispamový nástroj odesílatele A. V případě, že výzva bude vyhodnocena jako spam, tak se o ní uživatel A vůbec nedozví (případně dozví se zpožděním, které závisí na intenzitě návštěv jeho spamové složky) a původní email nedojde.

Pokud se existující antispamové nástroje nějakým způsobem upraví tak, aby odlišily výzvy od ostatních emailů, tak je víceméně jisté, že to bude předmětem zneužití ze strany spammerů.

- nedůvěra lidí v systém, kde může dojít ke ztracení důležitého mailu

Pro spoustu uživatelů bude přijatá výzva a v ní obsažená informace o „prozatímním nedoručení emailu příjemci“ znamenat příliš velké riziko nedoručení zprávy (co když se to někde ztratilo? opravdu to teď už bude doručeno? …)

- automaticky generované emaily

Systém může zamezit přijetí legitimních automaticky generovaných emailů, takže například velice stíží ba přímo znemožní registrace do emailových diskuzí, webové registrace za rozličným účelem atd. Pro každou takovou situaci bude muset uživatel vyvinou nadbytečné úsilí, aby problém vyřešil jinak (například existence jiného emailového účtu, který nepoužívá C-R)

- odesílatel používá unikátní adresu pro každý odeslaný email

Pokud systém odesílatele generuje pro každý odeslaný email unikátní adresu, tak bude pro každou generována výzva a navíc velikost whitelistu brzy přesáhne únosnou mez.

- přenesení antispamové zátěže na příjemce místo na spammera

Moderní a efektivní antispamové systémy by měly znepříjemňovat život těm, co spamy posílají a ne těm, co je přijímají. Navíc se tato metoda dotkne i poskytovatelů

internetového připojení (větší datová zátěž, místo jednoho emailu navíc výzva a odpověď na výzvu).

2.5 Greylisting

Metoda Greylisting je založena na myšlence, že každý normy [2] dodržující MTA[11] při neúspěšném pokusu o doručení emailu ho zařadí zpět do fronty a o doručení se pokusí později. Oproti tomu je dnes velká většina spamu odeslána specializovanými programy, které se nestarají o to, jestli byl email doručen, či nikoliv. Snaží se totiž rozeslat co nejvíce emailů za co nejkratší dobu a jelikož u spamu jde především o kvantitu, tak případné chyby během doručení se neřeší.

Greylisting je implementován na straně přijímacího MTA. Po přijetí emailu z neznámé adresy se uloží tzv. triplet, což IP adresa odesílajícího MTA a údaje ze SMTP obálky (adresa příjemce a odesílatele) a email se odmítne pomocí dočasné chyby (znamenající, že se má odesílající MTA pokusit o doručení znovu). Pokud dojde k opětovnému doručení emailu, tak ho systém pozná díky databázi uložených tripletů a v případě, že ještě neuplynul nastavený interval, tak email opět odmítne s dočasnou chybou, jinak email přijme a poznamená si, že pro uvedený triplet bude pro stanovený čas přijímat emaily bez omezení. Tato životnost tripletu je většinou v řádu týdnů a s větším počtem přijatých emailů obvykle roste, čímž se pro pravidelně komunikující strany zaručí plynulá komunikace.

Zřejmou nevýhodou Greylistingu je prvotní zpoždění u neznámých odesílatelů. Toto minimální zpoždění může být navíc větší, než je nastavený interval. Tato situace může nastat, pokud se na straně odesílatele o doručení pošty stará více MTA. Při dalším pokusu o doručení z jiného MTA se bude triplet lišit od předchozího pokusu (kvůli jiné adrese této MTA) a tím pádem bude brán tento pokus jako doručení jiného emailu. Tedy se uloží nový záznam do databáze tripletů.

Příklad: Systém odesílatele se skládá z jednoho hlavního a ze tří záložních MTA, které se postupně starají o doručení v případě předchozího neúspěšného pokusu. Nechť jsou tyto MTA postupně A,B,C a D a interval pro doručení na straně přijímací MTA (E) je 4 hodiny. Pomocí X jsou značeny zbylé dvě informace v tripletu, jenž se nemění.

[11] MTA - Mail Transfer Agent – program pro transport pošty

A se spojí s E, email je odmítnut, E si poznamená triplet (A,X,X). Za hodinu se pokusí o doručení B, email je odmítnut a E si zaznamená triplet (B,X,X). Za další 2 hodiny se pokusí o doručení C, email je odmítnut a E si zaznamená triplet (C,X,X). Zatím je vše v pořádku, jelikož od prvního pokusu o doručení uběhly 3 hodiny a to je méně, než nastavený interval 4 hodin na straně E. Co se ovšem stane, když se za další 2 hodiny pokusí o doručení D? Email je znovu odmítnut, přestože doba mezi prvním a tímto doručením (5 hodin) je větší, než nastavený interval (4 hodiny). Pokud se za další hodinu pokusí o doručení znovu C, tak je email zase odmítnut, jelikož doba mezi prvním doručením z C a tímto pokusem o doručení je 3 hodiny. V případě, že by se místo C o doručení pokusil B, tak by byl email přijat.

Doba doručení je tedy závislá na nastavení odesílajícího systému a nelze ji garantovat.

3 Návrh systému

V této kapitole je popsané vybrané řešení, dále přehled použitých součástí a vlastní návrh programu.

3.1 Zvolené řešení

S ohledem na uvažované možnosti uvedené v kapitole 1.3 byla nakonec zvolena jako nejvhodnější kombinace dvou antispamových metod a to konkrétně Bayesovského filtrování a Greylistingu. Původní idea byla ve využití myšlenky Challenge-response systémů místo Greylistingu, ale kvůli již uvedeným nedostatkům popsaným v kapitole 2.4 bylo od ní upuštěno[12].

Myšlenka nutnosti potvrzení komunikace ze strany příjemce se určitě nehodí jako samostatné řešení, lze ji ovšem smysluplně využít v mnoha situacích a proto byla do navrženého systému zakomponována.

Co se týká vlastního provedení návrhu systému, tak byla zvolena cesta ve využití již existujících nástrojů doplněných o vlastní část a to zejména z důvodu komplexnosti řešení, již ověřené funkcionality a tím pádem i určité důvěryhodnosti.

Systém je naprogramován v programovacích jazycích Java a PHP.

3.2 Použité součásti

Pro ohodnocení emailů na základě Bayesovkého skenování a inteligentní analýzy emailů byl vybrán program **jASEN**[13]. Jedná se o řešení, které je speciálně navržené pro integrování do stávajících antispamových systémů, jenž jsou naprogramovány v jazyce Java.

[12] I přes zmíněné nedostatky lze potenciál metody využít. Na konci práce je metoda uvedena jako jedno z možných rozšíření.
[13] Podle slov autorů [4] byl tento projekt nastartován právě na základě již zmiňovaného článku „A plan for Spam“ od Paula Grahama [7].

Jako poštovní server byl zvolen **qmail** a to zejména z důvodu jeho rozšířenosti. Samotná proxy je díky návrhu nezávislá na použitém poštovním serveru[14], její instalace se ovšem bude lišit.

Jako formát dat bylo zvoleno XML a pro manipulaci s ním Java API[15] **JAXB**, které slouží pro mapování XML struktury na Java třídy a naopak.

Pro logování bylo použito logovací API **log4j**.

Podrobnější popis programů jASEN, qmail, JAXB a log4j je uveden v příloze A.

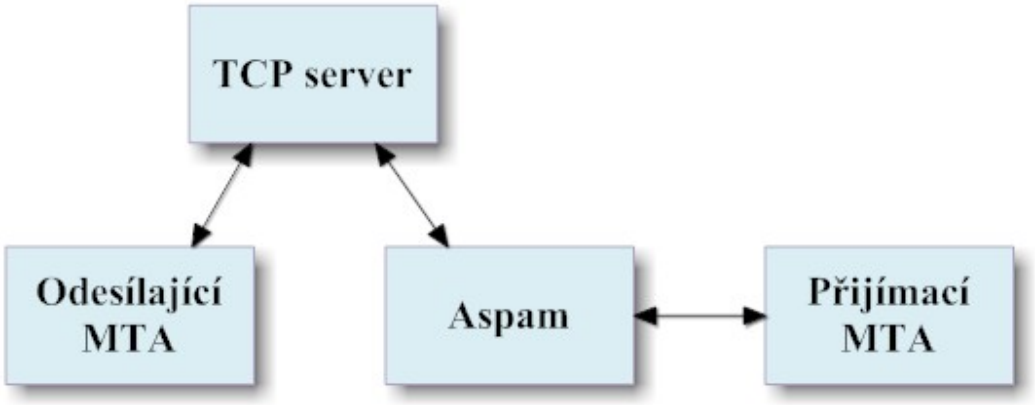
Nastavení programů jASEN, JAXB a log4j je uvedeno v příloze C.

3.3 Vlastní návrh programu

Navržený program Aspam funguje jako proxy, která se umístí mezi odesílající a přijímající MTA.

Odesílající MTA naváže prostřednictvím TCP serveru (program obsluhující příchozí spojení na portu 25) spojení s proxy a ta následně vstup analyzuje, vyhodnocuje a podle výsledku email přijme nebo prozatím odmítne. Odmítnutí je řešeno SMTP kódem, který informuje o dočasné chybě (4xx) a pro standardní SMTP server dodržující standardy RFC je signálem pro pozdější pokus o doručení zprávy. Proxy nefunguje jako samostatný SMTP server, ale komunikaci mezi oběma MTA předává a řídí, tudíž je řešení plně transparentní a zachová se původní nastavení a chování přijímajícího MTA.

[14] Platí za podmínky, kdy instalovaná MTA bude komunikovat přes standardní vstup a výstup a tedy lze proxy umístit mezi samotnou MTA a program obsluhující příchozí TCP spojení.
[15] Application Programming Interface - rozhraní pro programování aplikací.



Obrázek 1 - proxy Aspam

Pro analýzu emailu se používá program jASEN popsaný výše a na základě jeho ohodnocení je pro další krok použita následně modifikovaná metoda Greylisting:

- Rozhodnutí o přijetí či nepřijetí emailu neprobíhá ihned po přijetí SMTP obálky, tedy na základě tripletu, ale až po přijetí celého emailu a následném vyhodnocením programem jASEN.

- Podle tohoto ohodnocení se rozlišují různé hodnoty pro časový limit, který určuje za jak dlouho po prvním přijetí bude zpráva přijata (viz. popis metody Greylisting v kapitole 2.5) Tedy lze na základě ohodnocení značícího pravděpodobnost spamu urychlit přijmutí legitimní pošty a naopak pozdržet pravděpodobný spam.

- Nastavení hraničních hodnot pro Greylisting a uvedených hodnot časových limitů lze individuelně pro každý emailový účet.

- Pro záznamy v greylistu a whitelistu se místo tripletu používá adresa odesílatele a první IP adresa MTA uvedená v poli „Received:“, jenž se nachází v hlavičce emailu. Podle standartu RFC 2821 [2] pro SMTP protokol je každý server povinen při přijetí emailu přidat na začátek hlavičky emailu nový záznam „Received“, ve kterém budou uvedeny informace o odesílající MTA, včetně položky „from“ a v ní uvedené IP adrese odesílající MTA.

Pokud by byl záznam greylistu tvořen pouze emailovou adresou, tak by mohlo dojít ke snadnému překonání Greylistingu pomocí podvržené emailové adresy. I když nejsou adresy

zaznamenané ve whitelistu známy, tak v mnoha případech je lze uhodnout (například univerzálně používané názvy schránek jako je postmaster@..., atd.) Pokud se přidá další informace do záznamu, tak se uhodnutí podstatně minimalizuje. Navíc v případě posílání spamů jsou emaily rozesílány buď pokaždé z jiných adres, nebo v případě falšování údajů v hlavičce je podvrhnuta většina záznamů včetně této IP adresy. Tedy je velice nepravděpodobné, že by se dvojice (adresa odesílatele, IP adresa MTA) opakovala v případě posílání spamu.

Navíc se tímto eliminuje již zmiňovaná nevýhoda Greylistingu o větším minimálním zpožděním, než je nastavené (viz. kapitola 2.5). V případě odmítnutí a opětovných pokusů o doručení se nepřidávají žádné další informace do hlavičky a tudíž je identifikující dvojice (adresa odesílatele, IP adresa MTA) stále stejná, nezávisle na tom, který server se pokusil o doručení.

V případech, kdy se uživatel nechce spolehnout na Greylisting a možné zpoždění doručení pošty, lze nastavit, aby se v případě kladného ohodnocení emailu zaslal odesílateli informační email. V tomto emailu bude odkaz na webový formulář, kde může zájemce požádat o navázání komunikace.

Tento formulář zároveň slouží i jako další možná ochrana proti spamu, kde místo emailové adresy bude uživatel distribuovat webovou adresu a případní zájemci budou muset o komunikaci nejprve požádat. Po potvrzení od příjemce bude adresa přidána rovnou do whitelistu a tedy bude email doručen přímo.

4 Architektura

V následující kapitole je podrobněji popsána zvolená implementace programu, popis jeho součástí a hlavních algoritmů.

4.1 Přehled jednotlivých součástí

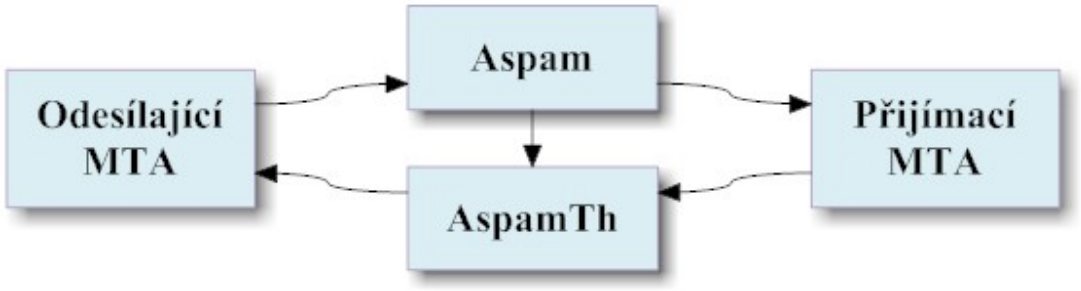
Navržený program se skládá z následujících částí:

- proxy Aspam - hlavní část, slouží pro samotný příjem pošty, řídí komunikaci mezi odesílající a přijímací MTA
- daemon AspamDaemon - zpracovává žádosti o komunikaci přes webový formulář
- webový formulář – slouží k podání žádosti o emailovou komunikaci

Proxy Aspam a daemon AspamDaemon jsou napsány v programovacím jazyku Java, webový formulář je napsaný v HTML a PHP.

4.2 Implementace proxy Aspam

Proxy Aspam se skládá ze dvou vláken, z nichž se každé stará o jeden směr komunikace. Program obsluhující příchozí spojení na portu 25 spustí po navázání komunikace s odesílající MTA proxy Aspam a na její standardní vstup bude předávat přijaté řádky od odesílající MTA. V hlavním vláknu proxy Aspam se spustí přijímací MTA a zároveň se vytvoří druhé vlákno vlastní proxy – **AspamTh**. Na standardní vstup vlákna AspamTh se přesměruje výstup spuštěné přijímající MTA a standardní výstup vlákna AspamTh se přesměruje na vstup odesílající MTA.



Obrázek 2 - komunikace mezi MTA

Aspam – Na standardní vstup dostává řádky od odesílající MTA a ty až do konce datové fáze SMTP přenosu ukládá a předává na standardní vstup přijímací MTA. Po načtení emailu, tj. po přijmutí řádky oznamující konec datové fáze[16], se email analyzuje a podle výsledku se buď odmítne, nebo přijme. Odmítnutí je řešeno tak, že se řádek oznamující konec datové fáze nepředá přijímací MTA, odesílající MTA dostane hlášení o chybě během zpracování[17] a spojení se ukončí. V případě přijmutí emailu se řádek oznamující konec datové fáze předá přijímající MTA a komunikace mezi MTA se dokončí.

AspamTh – na standardní vstup dostává řádky od přijímací MTA a ty hned předává odesílající MTA. Do tohoto směru komunikace se nezasahuje ani se přenášené řádky nikde neukládají.

4.2.1 Proces rozhodování o přijetí či nepřijetí emailu

Po přijetí emailu (konce datové fáze SMTP přenosu) probíhá následovně rozhodovací proces o přijetí/nepřijetí:

1. **krok** – test na přítomnost emailové adresy odesílatele a IP adresy MTA ve whitelistu

Pro splnění této podmínky a přijmutí emailu je potřeba:

- záznam je vytvořen pomocí AspamDaemonu a shoduje se emailová adresa, nebo

[16] <CRLF>.<CRLF>
[17] například: 451 local error in processing

- záznam je vytvořen pomocí jiného zdroje a shoduje se emailová adresa i IP adresa MTA

V případě nesplnění podmínky se pokračuje následujícím krokem.

2. krok – test na přítomnost emailové adresy odesílatele a IP adresy MTA v greylistu

Pro splnění této podmínky a přijmutí emailu je potřeba:

- shoduje se emailová adresa odesílatele, IP adresa MTA a zároveň je splněna časová prodleva, která určuje minimální dobu mezi přijetím emailu a prvním pokusem o jeho doručení (vytvoření tohoto záznamu v greylistu)

V případě pouhého nesplnění třetí podmínky, tj. časová prodleva není dostatečná, je email odmítnut, jinak se jedná o nově přijatý email z této dvojice adres (neexistuje záznam pro tuto dvojici adres v greylistu) a pokračuje se dalším krokem.

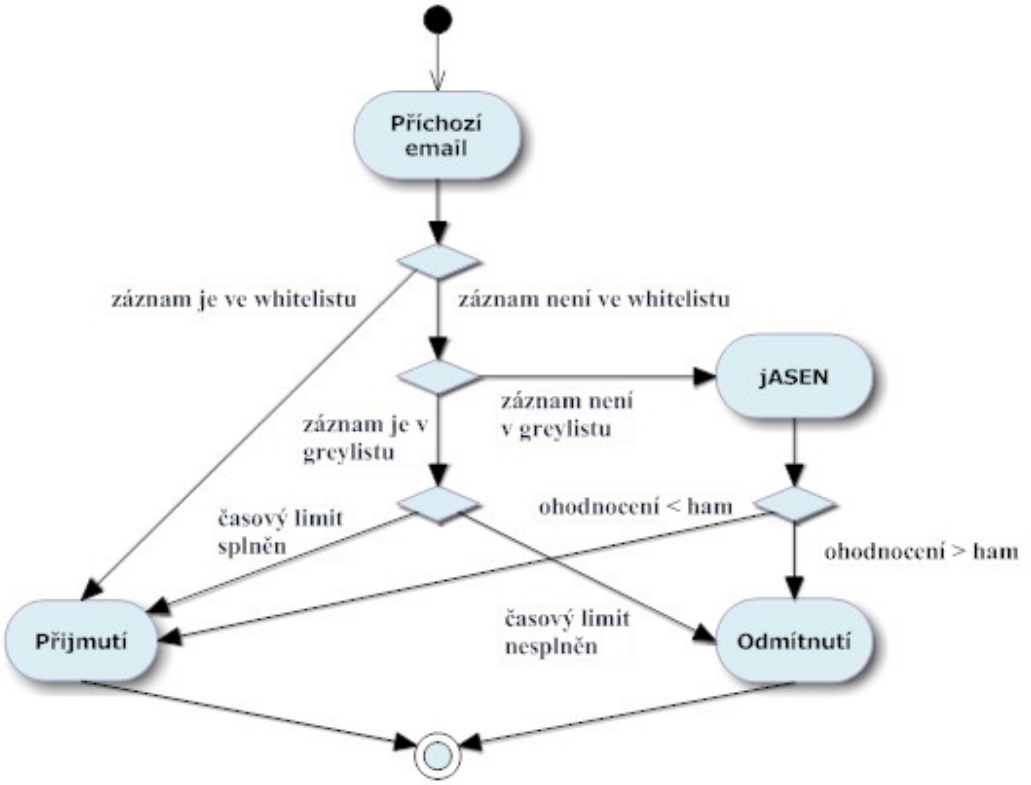
3. krok – testování pomocí programu jASEN

jASEN ohodnocuje emaily číslem v rozmezí <0,1>. Čím více se výsledek blíží maximální hodnotě, tím pravděpodobněji se jedná o spam. V závislosti na nastavení se rozlišují dvě hraniční hodnoty. První určuje, jaké musí mít email minimální ohodnocení, aby byl považován rovnou za spam, zatímco druhá určuje maximální ohodnocení pro nespam (ham).

Pro splnění této podmínky a přijmutí emailu je potřeba:

- ohodnocení emailu je nižší, než je stanovená hodnota pro nespam (ham)

V opačném případě je email odmítnut a vytvoří se nový záznam v greylistu.



Obrázek 3 - zpracování příchozího emailu

4.2.2 Formát dat

Pro uložení dat byl zvolen formát XML a to zejména pro snadnou a rychlou editovatelnost, přehlednost a existující javovské nástroje pro práci s XML.

Pro každou emailovou adresu příjemce se v podadresáři **data**[18] vytvoří XML soubor, který obsahuje následující informace:

- whitelist – seznam adres, ze kterých bude email přijat ihned bez testování
- greylist – seznam adres, ze kterých přišel email, ale nebyla splněna podmínka časového limitu

[18] jenž je umístěn v adresáři, kde je nainstalována proxy Aspam

- blacklist – seznam adres, ze kterých byl přijat email ohodnocený jako spam
- hraniční hodnoty pro spam a nespam – viz. kapitola 4.2.1
- časovou prodlevu při Greylistingu pro předpokládaný legitimní email a pro spam – viz. kapitola 3.3
- životnost záznamu v greylistu – po jakou dobu bude záznam uchován v greylistu
- nastavení, zda se má pro možný legitimní email (mající ohodnocení větší než maximální hodnota pro nespam a zároveň menší než hodnota pro spam) odeslat informující email – viz. kapitola 3.3

Pomocí JAXB se generují javovské třídy pro práci s těmito XML soubory.

Následující schéma XSD[19] popisuje strukturu, v níž jsou data uložena. V textu jsou rovněž uvedeny popisy některých struktur.

Schéma XSD:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

Každý soubor obsahuje whitelist, greylist, blacklist a nastavení pro konkrétního uživatele. Dále obsahuje informaci o vlastníkovi, datum vytvoření souboru, verzi a komentář.

```
<xsd:element name="wgblist" type="wgbListType"/>
<xsd:complexType name="wgbListType">
  <xsd:sequence>
  <xsd:element name="whitelist" type="whitelistType"/>
  <xsd:element name="greylist" type="greylistType"/>
  <xsd:element name="blacklist" type="blacklistType"/>
  <xsd:element name="settings" type="settingsType"/>
  </xsd:sequence>
```

[19] **XML Schema Definition**

```
  <xsd:attribute name="user" type="xsd:string"/>
  <xsd:attribute name="dateCreated" type="xsd:dateTime"/>
  <xsd:attribute name="version" type="xsd:string"/>
  <xsd:attribute name="comment" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="whitelistType">
  <xsd:sequence>
    <xsd:element name="address" type="addressW" minOccurs="1"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="comment" type="xsd:string"/>
</xsd:complexType>
```

Záznam pro whitelist obsahuje emailovou adresu odesílatele, IP adresu MTA, datum vytvoření a zdroj, jenž uvedený záznam vytvořil.

```
<xsd:complexType name="addressW">
  <xsd:sequence>
          <xsd:element name="from" type="xsd:string"/>
       <xsd:element name="ip" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="dateCreated" type="xsd:dateTime" use="required"/>
  <xsd:attribute name="source" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="greylistType">
  <xsd:sequence>
    <xsd:element name="address" type="addressBG" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="dateCreated" type="xsd:dateTime"/>
</xsd:complexType>

<xsd:complexType name="blacklistType">
  <xsd:sequence>
```

```
  <xsd:element name="address" type="addressBG" minOccurs="0"
maxOccurs="unbounded"/>
 </xsd:sequence>
 <xsd:attribute name="dateCreated" type="xsd:dateTime"/>
</xsd:complexType>
```

Společný formát adresy pro greylist a blacklist obsahuje emailovou adresu odesílatele, IP adresu MTA, ohodnocení emailu pomocí programu jASEN a datum vytvoření tohoto záznamu.

```
<xsd:complexType name="addressBG">
 <xsd:sequence>
  <xsd:element name="from" type="xsd:string"/>
  <xsd:element name="ip" type="xsd:string"/>
  <xsd:element name="jasen" type="xsd:double"/>
 </xsd:sequence>
 <xsd:attribute name="dateCreated" type="xsd:dateTime" use="required"/>
 </xsd:complexType>

<xsd:complexType name="settingsType">
 <xsd:sequence>
  <xsd:element name="greylistTime" type="greylistTimeType" minOccurs="1"
maxOccurs="1"/>
  <xsd:element name="greylistSpamTime" type="greylistTimeType" minOccurs="1"
maxOccurs="1"/>
  <xsd:element name="greylistTimeToLive" type="greylistTimeToLiveType"
minOccurs="1" maxOccurs="1"/>
```

Nastavení hraniční hodnoty pro předpokládaný spam.

```
  <xsd:element name="jasenSpamLevel">
   <xsd:simpleType>
    <xsd:restriction base="xsd:float">
        <xsd:minInclusive value="0"/>
        <xsd:maxInclusive value="1"/>
```

```
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
```

Nastavení hodnoty pro předpokládaný legitimní email.

```
<xsd:element name="jasenHamLevel">
    <xsd:simpleType>
      <xsd:restriction base="xsd:float">
          <xsd:minInclusive value="0"/>
          <xsd:maxInclusive value="1"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
```

Nastavení emailové notifikace pro předpokládaný legitimní email

```
  <xsd:element name="HamLevelNotification" type="xsd:boolean"/>

 </xsd:sequence>
</xsd:complexType>
```

Nastavení časové prodlevy, jenž je nezbytná pro přijetí emailu od prvního pokusu o jeho doručení.

```
<xsd:complexType name="greylistTimeType">
 <xsd:sequence>
  <xsd:element name="hours" type="xsd:byte"/>
  <xsd:element name="minutes" type="xsd:byte"/>
 </xsd:sequence>
</xsd:complexType>
```

Nastavení životnosti záznamu v greylistu, po jejímž uplynutí bude tento záznam z greylistu odstraněn. Slouží k odstranění starých a nepotřebných záznamů a tedy k zachování rozumné velikosti seznamu.

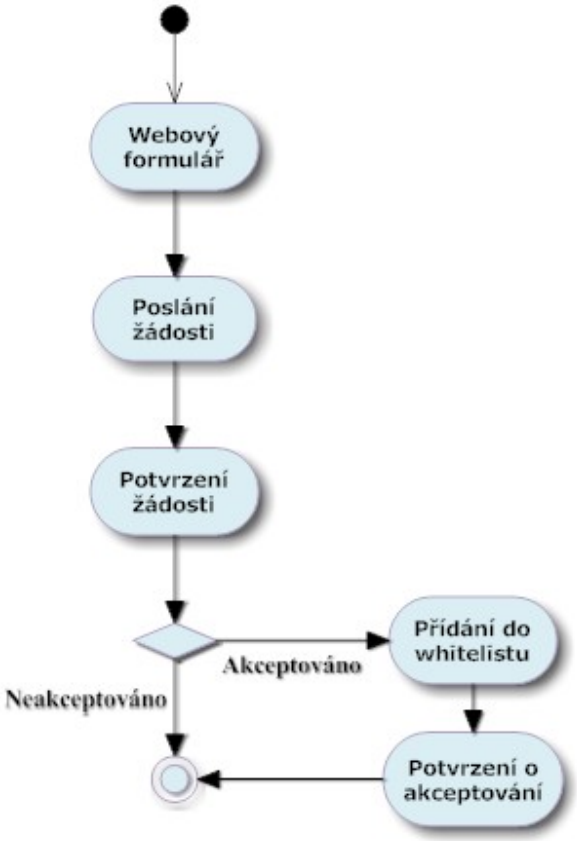
```
<xsd:complexType name="greylistTimeToLiveType">
  <xsd:sequence>
    <xsd:element name="days" type="xsd:byte"/>
    <xsd:element name="hours" type="xsd:byte"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

4.3 Žádost o emailovou komunikaci

Navržený systém umožňuje na žádost zájemce o emailovou komunikaci přidat jeho emailovou adresu do whitelistu příjemce.

Zájemce vyplní webový formulář, kde uvede svoje jméno, email a volitelně poznámku (dodatečné informace pro příjemce). Po odeslání formuláře se pošle příjemci na jeho adresu email, kde jsou uvedené informace a též webový odkaz, kterým tuto žádost může potvrdit. Po potvrzení se žadatelova adresa přidá do whitelistu a zároveň mu bude zaslán potvrzující email.

O tuto část se stará program AspamDaemon, jehož popis a funkce jsou uvedeny v následující kapitole.



Obrázek 4 - zpracování žádosti o komunikaci

4.4 Implementace daemonu AspamDaemon

4.4.1 Funkce AspamDaemonu

AspamDemon je daemon, který ve zvoleném intervalu:

- kontroluje obsah adresáře, do kterého se ukládají přijaté žádosti o komunikaci a pro každou zde nalezenou žádost odešle informační email příjemci. Informační email obsahuje položky vyplněného formuláře a dále odkaz, kterým lze potvrdit tuto žádost.

- kontroluje obsah adresáře, do kterého se ukládají žádosti čekající na potvrzení a vymaže zde všechny nalezené žádosti, jenž jsou starší než nastavená životnost žádostí.

- kontroluje obsah adresáře, do kterého se ukládají akceptované žádosti o komunikaci a každou zde nalezenou žádost zpracuje, tedy přidá emailovou adresu žadatele do whitelistu příjemce a zároveň odešle potvrzující email žadateli, že jeho žádost byla akceptována.

AspamDaemon se spouští s následujícími parametry:

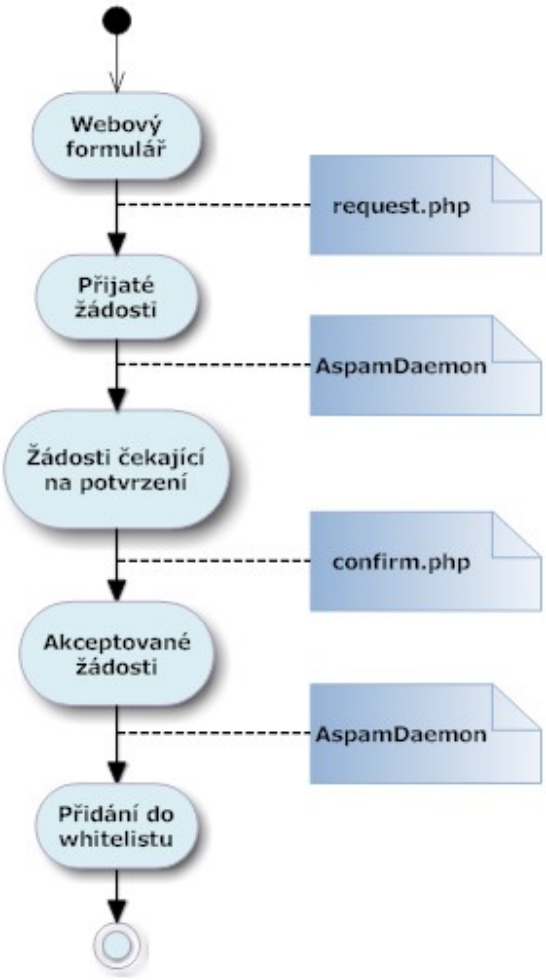
- interval spouštění (v sekundách)
- životnost žádostí čekajících na potvrzení (v hodinách)
- cesta k adresáři pro přijaté žádosti
- cesta k adresáři pro žádosti čekající na potvrzení
- cesta k adresáři pro potvrzené žádosti

4.4.2 Postup zpracování žádosti

Uživatel si načte webovou stránku, která bude obsahovat žádost o komunikaci v podobě formuláře, do kterého uvede svoje jméno, emailovou adresu a volitelně komentář.

Po odeslání vyplněného formuláře se do adresáře pro přijaté žádosti pomocí PHP vygeneruje soubor, který bude obsahovat emailovou adresu příjemce (nastavená v PHP skriptu) a dále položky z formuláře. AspamDaemon zde při spuštění tento soubor najde, pošle oznámení o žádosti na emailovou adresu příjemce a soubor přesune do adresáře pro žádosti čekající na potvrzení.

Potvrzení žádosti proběhne kliknutím na odkaz uvedený v oznámení. Ve vygenerovaném odkazu je adresa PHP skriptu a jméno souboru, který tento skript přesune do adresáře pro potvrzené žádosti. AspamDaemon zde při spuštění tento soubor najde, přidá adresu žadatele do whitelistu příjemce a odešle žadateli oznámení o přijetí jeho žádosti. Tímto se zároveň žadatel dozví emailovou adresu příjemce.



Obrázek 5 - jednotlivé kroky zpracování žádosti

5 Zhodnocení práce

5.1 Dosažené cíle

V navrženém systému byla použita kombinace několika metod boje proti spamu. Toto řešení obecně dává celkově lepší výsledky, než při použití jedné metody. Každá metoda má svoje slabé stránky, které lze při troše snahy obejít a proto lze nedostatky jedné metody eliminovat současným použitím jiné metody.

Hlavní nevýhoda metody Greylisting, tedy prvotní zpoždění přijetí emailu z neznámé adresy, byla zmírněna pomocí Bayesovského ohodnocení emailu, jenž používá systém jASEN. Předpokládaná legitimní zpráva tedy bude přijata ihned, narozdíl od standardní implementace metody Greylisting.

Další rozdíl oproti standardnímu Greylistingu je rozhodovací fáze, která se neprovádí ihned po přijetí tripletu, ale až po přijetí celého emailu. Výhoda je v tom, že se tímto metoda odlišuje od ostatních implementací Greylistingu, tudíž pokud se jednou spammeři přizpůsobí chybovému hlášení po odeslání tripletu[20], tak to v případě uvedeného řešení nebude mít efekt. Další výhodou je mnohem více informací, podle kterých lze rozhodovat, tedy jaké informace budou v záznamech greylistu. Mezi nevýhody tohoto způsobu samozřejmě patří vyšší datová zátěž oproti samotnému Greylistingu, jelikož zde se musí email pokaždé přijmout celý.

Jako smysluplná se též jeví implementovaná možnost rozdílných časových limitů pro Greylisting a to na základě ohodnocení emailů. Pro předpokládané spamy lze nastavit mnohem větší limit pro přijetí, čímž by se dal eliminovat pokus o doručení za delší dobu, třeba za účelem spammera obejít instalovaný standardní greylist, u kterého jsou nastavené limity v rámci hodin. Tento pokus o opětovné doručení by z pohledu spammera zase neměl být za delší dobu, než je pár hodin, jelikož je docela pravděpodobné, že jeho adresa už bude na nějakém veřejném blacklistu, nebo v případě rozeslání spamu z napadeného poštovního serveru bude administrátor o zneužití informován a problém vyřeší.

[20] což je v případě většího rozšíření antispamových metod používajících Greylisting téměř jisté

Nevýhody samotného Bayesovského ohodnocení, mezi které zvláště poslední dobou patří obrázkové spamy, jsou řešeny pomocí různých modulů programu jASEN (popsáno v kapitole A.2).

Ten zde hraje zásadní roli, jelikož veškeré analýzy a rozhodování probíhají na základě jeho ohodnocení. Pro dobré výsledky je tedy nezbytně nutné udržovat databázi spamů a hamů, na které se bude trénovat Bayesův filtr.

V případě nesprávného ohodnocení, tj. velmi kladného ohodnocení nevyžádané zprávy, se emailová adresa přidá do whitelistu a tedy bude i doručena. Jelikož ale většina spamů chodí pokaždé z jiné adresy[21] a navíc záznam ve whitelistu tvoří kromě emailové adresy i IP adresa MTA, tak se nejedná o zásadní nevýhodu. Pravdou ovšem je, že tímto spam překoná všechny další implementované ochrany, které by v případě o něco horšího ohodnocení (kde by už email nespadal do kategorie předpokládaný nespam) byly začleněny do procesu rozhodování. Zde je tedy otázka, jestli je přijatelnější občasné doručení nevyžádané zprávy, nebo třeba aplikace Greylistingu na všechny nově příchozí emaily bez výjimky a tedy i určitého zpoždění pro legitimní emaily.

Celkově má program větší náročnost na systémové zdroje a to zejména díky použití programu jASEN na analýzu zpráv. Nehodí se tedy pro nasazení do systémů, jenž jsou určeny pro velké množství současně zpracovávaných zpráv.

5.2 Možnosti budoucího rozšíření

5.2.1 Challenge response

I přes jmenované nevýhody Challenge-response systémů by stálo za úvahu, jestli tuto metodu není výhodné implementovat pro část emailů splňujících jisté ohodnocení. Například ji lze aplikovat jako možnou variantu k posílání upozornění o zpoždění (kap. 3.3), kdy po přijetí odpovědi na výzvu bude původní email přijat ihned.

[21] nezávisle na tom, jestli je tato adresa pravdivá, nebo podvržená.

5.2.2 Zpomalení SMTP komunikace

Jenda z dalších antispamových metod je záměrné zpomalení komunikace. Tuto metodu nelze použít obecně, jak už bylo popsáno v kapitole 2.2. Na druhou stranu lze tento princip aplikovat i za stávajícího protokolu pro výměnu elektronické pošty a to například pro skupinu emailů, jejíchž ohodnocení programem jASEN spadá do určitého intervalu. Na tyto emaily lze použít následující formu zpoždění:

Při opětovném doručení emailu, pro který už je záznam v greylistu, lze pozdržet odpověď o přijetí emailu (250 OK). Maximální doba mezi začátkem přenosu dat a jeho úspěšným ukončením je podle standartu RFC 10 minut, což už je relativně dlouhé zdržení pro systém snažící se o poslání co nejvíce emailů v co nejkratším čase.

5.2.3 Implementace blacklistu

Součásti navrženého řešení je částečně i blacklist, který je uveden v XML struktuře dat, existují funkce pro operaci s ním, ale momentálně není použit v rozhodovací časti programu. Jelikož se jedná o výběr vyžádaných zpráv z celkového toku přijímaných zpráv, tak případné zahození emailu kvůli přítomnosti adresy v blacklistu je mimo rámec zadání. Většinou se ovšem vyplatí nějakou formu blacklistu použít a proto je tedy částečně implementován.

5.3 Závěr

Problematika spamu je velice rozsáhlá a boj jenž je proti němu veden zatím ne příliš úspěšný. Zatím stále platí, že jsou rozesílatelé spamů vždy o krok napřed a vyskytne-li se nějaké nové „převratné“ antispamové řešení, tak se mu brzo dokáží přizpůsobit.

Otázkou je, proč je spam i přes snahy o jeho omezení pořád výnosný, když ho přeci „každý ignoruje a ihned maže“. Bohužel tohle evidentně není pravda a pořád se najde dost jedinců, kteří na spamem inzerované zboží odpoví, nechají se zlákat super výhodnou nabídkou čehokoliv, či jenom naivně věří, že se na ně usmálo štěstí a opravdu vyhráli slibovaný obnos.

Všechno[22] v dnešním světě se točí kolem peněz a dokud bude nějaký statek či služba vydělávat, tak se bude provozovat, nehledě na případné překážky či morální zásady.

Obecně platí, že pokud se podaří snížit výnosy inzerentů pod hranici nákladů, tak podnikání v tomto odvětví přestane být rentabilní. Rozesílání spamu představuje byť minimální, tak přeci jenom nějaké náklady, které se musí inzerentovi či zadavateli spamu vrátit.

Konečným cílem všech metod a postupů by tedy hlavně mělo být snížit výnosy inzerentů a zároveň zvýšit jejich náklady tak, aby se prostě přestalo vyplácet spamy posílat.

[22] názor autora

Literatura

[1] STRATO, http://www.strato-hosting.co.uk/press/press/2006/2007_2_6.html

[2] RFC 2821, http://www.ietf.org/rfc/rfc2821.txt

[3] qmail, http://www.qmail.org/top.html

[4] jASEN - The pure java Anti Spam ENgine, http://www.jasen.org/

[5] Chi-square distribution, http://en.wikipedia.org/wiki/Chi-square_distribution

[6] JAXB, https://jaxb.dev.java.net/

[7] Paul Graham – A Plan for Spam, http://www.paulgraham.com/spam.html

Seznam obrázků

A Použité produkty třetích stran

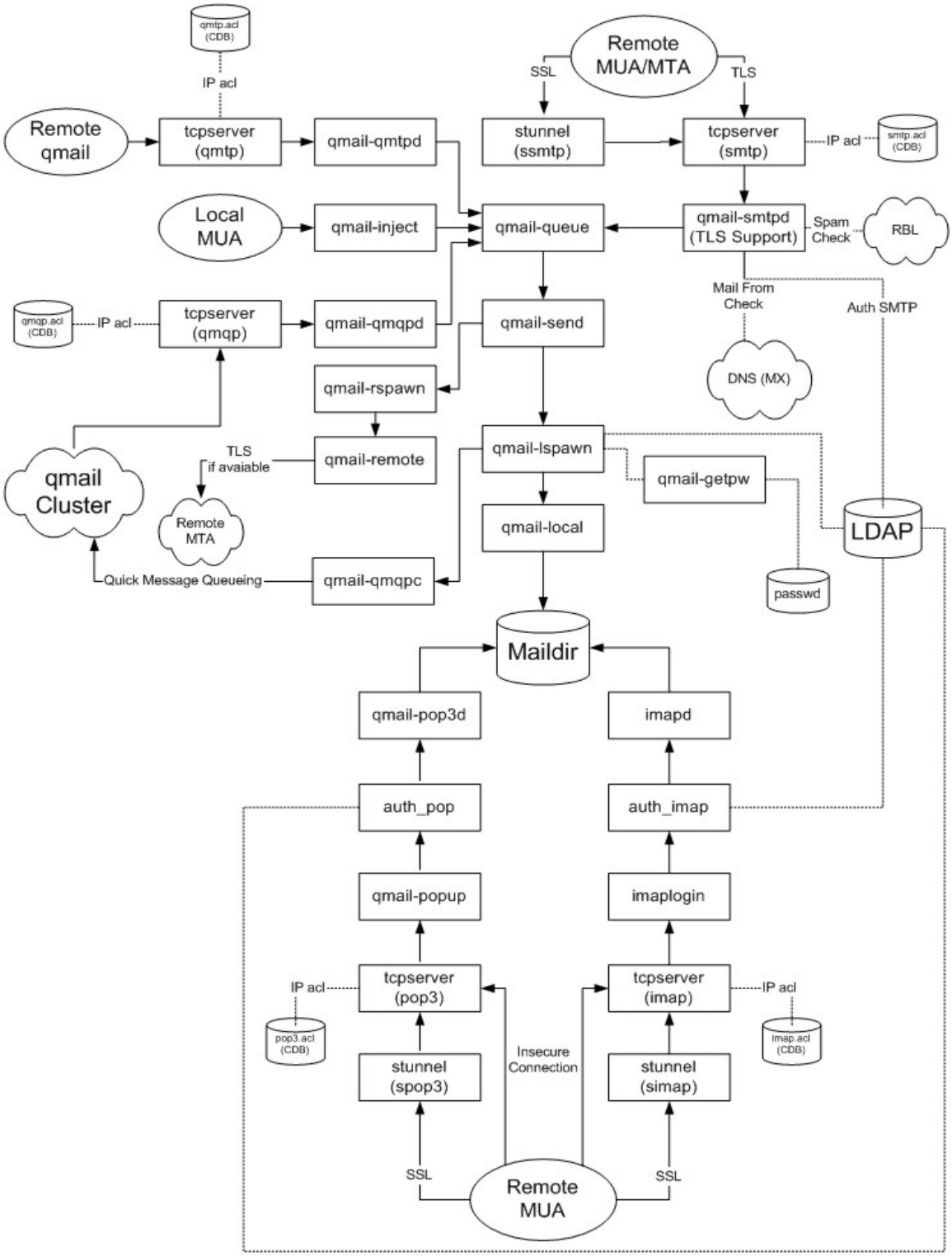
V následujícím přehledu je uveden popis programů třetích stran, které jsou součástí navrženého řešení a výsledný systém je používá.

A.1 qmail

qmail [3] je dnes druhý nejrozšířenější MTA server na internetu. Celý program je tvořen řadou modulů tvořících řetězec, kterým prostupuje každá zpráva. Jednotlivé moduly jsou navrženy tak, aby byly pokud možno jednoúčelové a mohly běžet s minimálními nutnými uživatelskými právy, tudíž je zde kladen důraz hlavně na bezpečnost, rychlost a spolehlivost.

Uvedená modularita umožňuje snadné rozšíření, což je důvodem, proč byl qmail zvolen jako součást této implementace.

Schéma fungování qmailu je zobrazeno na obrázku č.6



Obrázek 6 - schéma MTA qmail

A.2 jASEN

jASEN [4] je antispamový nástroj kompletně napsaný v programovacím jazyku Java kombinující Bayesovské skenování emailů s inteligentní analýzou jejich obsahu podle různých charakteristik.

K ohodnocení emailů se používají následující moduly:

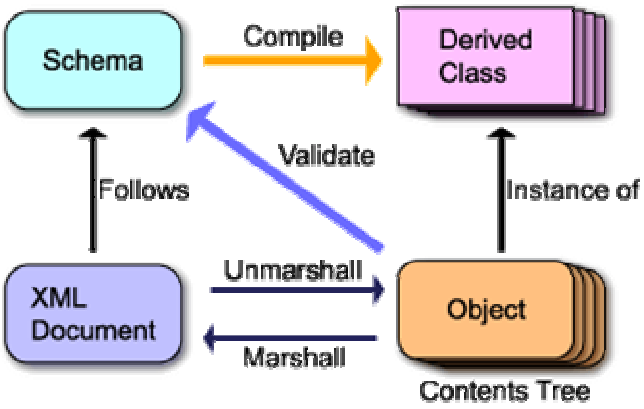
- AnomalousCharacterScanner – hledá nestandardní znaky, které ve větším množství často indikují SPAM
- AttachmentScanner – analyzuje přílohu emailu na výskyt nestandardních nebo nebezpečných typů souborů
- FromAddressValidationScanner – kontroluje From hlavičku proti odesílateli obálky a zpáteční adrese
- HeuristicScanner – heuristická analýza na typické příznaky spamu
- HTMLConcealmentScanner – detekuje maskování výrazů pomocí HTML
- ImageDominanceScanner – analyzuje poměr obrázků vůči textu, typické zejména pro obrázkové spamy
- InvisiMailScanner – indikuje emaily bez obsahu a bez předmětu zprávy
- KeywordScanner – test na výskyt typicky spamových výrazů
- ObfuscatedCharacterScanner – test na výskyt znaků reprezentující jiné znaky (např. SP@M)
- RBLScanner – kontrola odesílajícího serveru oproti RBL[23] databázi
- RecipientScanner – kontrola nadměrného počtu příjemců emailu

[23] Realtime Blackhole List

- RobinsonScanner – pravděpodobnostní scanner, jenž používá chi-square [5] verzi Bayesova klasifikátoru
- SenderAddressValidationScanner – kontroluje podvrhnutou adresu odesílajícího serveru
- TagFalseAnchorScanner – kontrola odkazů na rozdílnost href atributu při zobrazení oproti zápisu ve zdrojovém kódu
- TagSourceCgiScanner – detekce obrázků odkazujících na CGI skript pro jejich zobrazení
- TagSourcePortScanner – detekce nestandardních portů v URL odkazech

A.3 JAXB

JAXB[24] [6] slouží pro mapování XML struktury na Java třídy a naopak. Umožňuje data z XML dokumentu namapovat do objektu, s touto instancí třídy poté libovolně pracovat a nakonec objekt uložit jako původní XML dokument.



Obrázek 7 - schéma JAXB

[24] Java Architecture for XML Binding

A.4 Log4j

Log4j je logovací API pro Javu, které umožňuje hierarchické logování a podporuje několik formátů výstupu.

B Obsah přiloženého CD a popis zdrojových kódů

B.1 Aspam, AspamDaemon

V adresáři **Aspam/src** se nacházejí uvedené javovské balíky:

`cz.cuni.mff.antispam` – hlavní funkční část proxy Aspam, obsahuje následující třídy:

- `AnalyzeHeader` – analyzuje hlavičku emailu, ze které získá adresu příjemce, adresu odesílatele a IP adresu MTA uvedenou v poli „Received:“.

- `AnalyzeMessage` – analyzuje celou zprávu, získané údaje z hlavičky a SMTP obálky porovnává se záznamy ve whitelistu a greylistu, ohodnocuje zprávu pomocí API jASEN, přidává záznamy do whitelistu a greylistu a odesílá notifikační email pro kladně ohodnocenou zprávu v případě, je-li nastaven.

- `Aspam` – řídí komunikaci od odesílající MTA k přijímací MTA, po přijetí řádky označující konec datové fáze předá zprávu k vyhodnocení třídě AnalyzeMessage a podle výsledku komunikaci dokončí, nebo ukončí posláním dočasné chyby.

- `AspamTh` – přeposílá komunikaci od přijímací MTA k odesílající MTA.

- `FileManager` – diskové operace s uloženými daty. Vytvoření, zápis a čtení souborů ve formátu XML, včetně mapování XML dat na Java třídy a naopak.

- `WGBList` – funkce pro přidání a vyhledání záznamů ve whitelistu, greylistu a blacklistu, vytvoření záznamů pro nové adresy včetně nastavení defaultních hodnot jednotlivých parametrů.

`cz.cuni.mff.antispam.generate` – vygenerované třídy pomocí JAXB, operace s javovskými objekty, které obsahují namapovaná data z XML dat.

`cz.cuni.mff.antispam.daemon` – funkční část daemonu AspamDaemon, obsahuje třídu `AspamDaemon`.

`cz.cuni.mff.antispam.messages` – nastavení používaných zpráv v programu, obsahuje třídu `Messages` pro výběr nastavených hodnot, vlastní data jsou uloženy v souboru `Messages.properties`

V adresáři **Aspam** je schéma `wgblist.xsd` používaného formátu XML dat (viz. 4.2.2) a též ant skript `build.xml` pro vygenerování javovských tříd pomocí tohoto schématu.

V adresáři **Aspam/bin** jsou přeložené třídy z adresáře **Aspam/src**.

Do adresáře **Aspam/data** se ukládají data vlastní aplikace, tedy XML soubory pro jednotlivé emailové účty.

V adresáři **Aspam/doc** je umístěna programová dokumentace.

V adresáři **Aspam/lib** jsou potřebné knihovny pro používané programy i pro běh samotné proxy a daemonu. Dále jsou zde data a nastavení pro program jASEN.

V adresáři **Aspam/log4j** je nastavení pro log4j.

V adresáři **Aspam/supervise** je umístěn vzorový skript pro spouštění AspamDaemonu pomocí daemontools (viz. C.2.2)

V adresáři **install** je zabalená distribuce určená k instalaci (viz. C.2)

B.2 Webový formulář

V adresáři **web** se nachází následující dva PHP skripty

- `request.php` – obsahuje webový formulář, po jehož odeslání se vytvoří v adresáři pro přijaté žádosti náhodně vygenerovaný soubor, jenž bude obsahovat na prvním řádku adresu příjemce, na druhém řádku adresu žadatele, na třetím řádku uvedené jméno a ve zbytku souboru bude poznámka.

- `confirm.php` – jako parametr dostane název souboru obsahující žádost, která má být potvrzena. Pokud tento soubor existuje v adresáři pro žádosti čekající na potvrzení, tak je emailová adresa žadatele (na druhém řádku) přidána do whitelistu příjemce (první řádek).

C Návod k instalaci

C.1 MTA qmail

Instalace poštovního serveru je relativně náročná a kompletní popis všech kroků a nastavení velice obsáhlý, proto je zde uveden pouze odkaz na výborné stránky, jenž se instalací a nastavením zabývají.

http://www.lifewithqmail.org/

C.2 Aspam

Pro fungování programu je třeba mít na cílovém místě nainstalovanou Javu. Popisovaná instalace je určena pro verzi 1.6 [25].

V adresáři **install** na přiloženém cd se nachází soubor `aspam.tar.gz`. Ten se nakopíruje na požadované místo pro instalaci programu a rozbalí pomocí příkazů

gunzip `aspam.tar.gz`
tar –xpf `aspam.tar`

C.2.1 Proxy Aspam

V případě instalace qmailu podle bodu C.1 se změní supervise skript `run`, jenž je umístěný v adresáři **/var/qmail/supervise/qmail-smtpd/**

Stávající příkaz *exec* se zamění za následující dva příkazy:

cd $ASPAMPATH
*exec /usr/local/bin/tcpserver -v -R -l "$LOCAL" -x /etc/tcp.smtp.cdb -c "$MAXSMTPD" *

[25] pro nižší verze Javy je nutné ve spouštěcím příkaze *java* vypsat za parametrem *–cp* všechny použité knihovny zvlášť.

-u "$QMAILDUID" -g "$NOFILESGID" 0 smtp java -cp $ASPAMPATH/bin:$ASPAMPATH/lib/jasen_lib/:$ASPAMPATH/lib/jaxb_lib/*:$ASPAMPATH/lib/* cz/cuni/mff/antispam/Aspam /var/qmail/bin/qmail-smtpd 2>&1*

a zároveň se nastaví cesta k aktuálnímu umístění programu Aspam do nové proměnné ASPAMPATH (např. /usr/aspam).

Hodnota $MAXSMTPD znamená maximální počet souběžně přijímaných emailů a tedy by se měla volit s ohledem na parametry serveru (zejména množství dostupné paměti).
Dále je potřeba pro podadresář **data** nastavit práva pro zápis od uživatele *qmaild*, nebo rovnou změnit vlastníka tohoto adresáře na uživatele *qmaild*.

C.2.2 Daemon AspamDaemon

pro Aspam daemon se musí vytvořit následující supervise skript:

```
#!/bin/sh
ASPAMPATH= XXX
TIMEPERIOD= XXX
LIFETIME= XXX
REQUESTED= XXX
SENT= XXX
CONFIRMED= XXX
cd $ASPAMPATH
/usr/local/bin/setuidgid qmaild java -cp
$ASPAMPATH/bin:$ASPAMPATH/lib/jaxb_lib/*:$ASPAMPATH/lib/jasen_lib/log4j-
1.2.9.jar:$ASPAMPATH/lib/* cz/cuni/mff/antispam/daemon/AspamDaemon
"$TIMEPERIOD" "$LIFETIME" "$REQUESTED" "$SENT" "$CONFIRMED"
```

Nastavení jednotlivých parametrů:

ASPAMPATH	cesta k aktuálnímu umístění programu Aspam
TIMEPERIOD	doba v sekundách určující jak často se bude daemon pouštět
LIFETIME	doba v hodinách určující životnost žádostí čekajících na potvrzení
REQUESTED	adresář, do kterého se budou ukládat přijaté žádosti
SENT	adresář, do kterého se budou ukládat žádosti čekající na potvrzení od příjemce

CONFIRMED adresář, do kterého se budou ukládat potvrzené žádosti čekající na zpracování

Uvedené tři adresáře se musí zároveň vytvořit a nastavit práva pro zápis od uživatele *qmaild*, nebo rovnou změnit vlastníka těchto adresářů na uživatele *qmaild*.

Skript musí být spustitelný (*chmod 775*) a zároveň se musí vytvořit link na adresář obsahující tento skript do adresáře **/service** (*ln -s*)

C.2.3 Nastavení proxy Aspam

Nastavení všech textů je v souboru `Messages.properties`, jenž je uložen v adresáři **./bin/cz/cuni/mff/antispam/messages**

Význam jednotlivých položek nastavení:

- server.name – jméno serveru, na kterém Aspam běží. Pro zprávy odesílané z tohoto serveru se bude proxy jevit jako neaktivní.
- SMTP.return.command – hlášení o chybě, které se odešle odesílající MTA v případě odmítnutí emailu.
- Confimation.subject – předmět zprávy informující o akceptované žádosti o komunikaci
- Confimation.message – text zprávy informující o akceptované žádosti o komunikaci
- Request.link – webová adresa skriptu, kterým se akceptuje žádost
- Request.subject - předmět zprávy informující o přijaté žádosti o komunikaci
- Request.message - text zprávy informující o přijaté žádosti o komunikaci
- Request.from – uvozující text pro jméno žadatele
- Request.email - uvozující text pro emailovou adresu žadatele
- Request.comment - uvozující text pro komentář žadatele
- Notification.subject - předmět zprávy informující o pozdržení emailu díky antispamovému řešení

- Notification.message – text zprávy informující o pozdržení emailu díky antispamovému řešení
- Notification.from – adresa uvedená v poli „from:“ ve zprávě informující o pozdržení emailu díky antispamovému řešení

C.3 Nastavení skriptů PHP

Ve skriptu `request.php` se do proměnné **myEmail** nastaví email uživatele, pro kterého je žádost podána a do proměnné **pathRequested** se nastaví cesta k adresáři pro přijaté žádosti.

Ve skriptu `confirm.php` se do proměnné **pathSent** nastaví cesta k adresáři pro žádosti čekající na potvrzení a do proměnné **pathConfirmed** se nastaví cesta k adresáři pro potvrzené žádosti.

C.4 Defaultní hodnoty

V java třídě WGBlist, která má na starost vytvoření nového záznamu pro každý emailový účet na serveru, jsou nastavené následující defaultní hodnoty:

- časový limit Greylistingu je 1 hodina pro předpokládaný legitimní email a 12 hodin pro pravděpodobný spam
- životnost záznamů v greylistu je 2 dny a 12 hodin
- hraniční hodnota pro spam je 0.8, hraniční hodnota pro ham je 0.1
- emailová notifikace v případě emailu ohodnoceného v rozmezí <0,1 , 0,8> je nastavena na false (vypnuto)

C.5 Generování javovských tříd pro práci s XML daty

Javovské třídy se generují pomocí JAXB spuštěním ant skriptu `build.xml`, jenž je umístěn v adresáři **Aspam**. Popis schématu XSD, z něhož se uvedené třídy generují, je uveden v kapitole 4.2.2.

C.6 Nastavení log4j

V adresáři **Aspam/log4j** se nachází soubor `log4j.properties`, ve kterém je následující nastavení jednotlivých logů.

Kořenový logger, loguje vše na úrovni WARN a výše do souboru `aspam.log`.

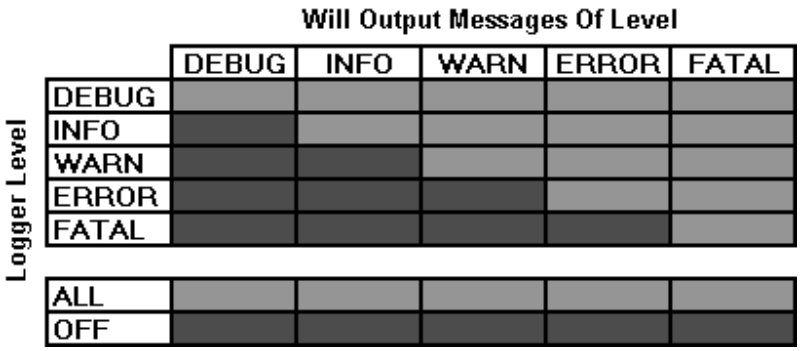
log4j.rootLogger=WARN, R
log4j.appender.R.File=aspam.log

Informační log pro proxy Aspam, slouží pro základní přehled o přijímaných emailech a loguje vše na úrovni INFO a výše.

log4j.logger.log2=INFO, T
log4j.appender.T.File=aspam2.log

Informační log pro daemon AspamDaemon, loguje vše na úrovni INFO a výše.

log4j.logger.log3=INFO, D
log4j.appender.D.File=aspamdaemon.log



Obrázek 8 - logovací úrovně Log4j

C.7 Nastavení programu jASEN

Nastavení k jednotlivým modulům programu jASEN je v podadresáři **lib/jasen-conf**. Význam jednotlivých nastavení je podrobně popsán na

http://www.jasen.org/configuration.php