Ironmail – Filtrando emails indesejados

Leonard Bohrer Spencer, Guilherme Bertoni Machado

FATEC RS – Faculdade de Tecnologia Senac RS
Coronel Genuíno 130 – Cidade Baixa – 90.001-350 – Porto Alegre – RS – Brasil
lbohrer@terra.com.br, gb.machado@sinprors.org.br

Abstract. *This article aims to present the Ironmail, a tool capable of eliminating a large amount of unwanted emails that are delivered in mailboxes every day. Spams, viruses, spyware and other problems normally contained in these emails can be blocked simply by using some policies for sending and receiving emails, antispam and antivirus tools, all working together. Through the functional requirements analysis the Ironmail was modeled to offer a number of advantages, as policies on sending and higher performance, compared to other solutions found in the literature and in the market.*

Resumo. *Este artigo tem por objetivo apresentar o Ironmail, uma ferramenta capaz de eliminar uma grande quantidade de emails indesejados que são entregues nas caixas-postais todos os dias. Spams, vírus, spywares e outros problemas normalmente oriundos destes emails podem ser bloqueados utilizando simplesmente algumas políticas para envio e recebimento, um antispam e um antivírus, todos trabalhando em conjunto. Através da análise dos requisitos funcionais foi possível modelar o Ironmail para que ele oferecesse uma série de vantagens, como políticas de envio por usuário e maior desempenho, em relação às demais soluções encontradas na literatura e no mercado.*

1. Introdução

Simultaneamente ao desenvolvimento e popularização da Internet ocorreu o crescimento de um fenômeno que, desde seu surgimento, se tornou um dos principais problemas da comunicação eletrônica em geral: o envio em massa de mensagens não solicitadas. Esse fenômeno ficou conhecido como spamming, as mensagens em si como *spam* e seus autores como *spammers* [Antispam.br 2007].

Embora algumas leis a respeito do assunto já tenham sido aprovadas, ainda não existe uma legislação definitiva que regule a prática do *spamming* ou a caracterize como sendo crime. Apesar desta atual indefinição legal, diversas entidades governamentais, comerciais e independentes declaram que o spam é um dos maiores problemas atuais da comunicação eletrônica.

Para controlar o tráfego sobre o conteúdo dos *emails* e assim poder evitar entrada de *spams*, vírus, *trojan horses* e outros, existem algumas ferramentas, como o Amavisd-new e o Mailscanner, relatados na seção 3, mas de difícil instalação e faltando alguns requisitos necessários para um bom controle sobre o servidor como, por exemplo, políticas por usuário para envio de emails. Estas ferramentas são instaladas no servidor de *emails* para efetuar a filtragem antes mesmo de chegar ao cliente de *emails*.

O Ironmail é uma ferramenta para auxiliar na segurança de redes de computadores e tem por objetivo proteger o servidor de *emails* e estações de trabalho. O objetivo é efetuar o controle total sobre os *emails* enviados e recebidos através da diminuição no volume de *spams* utilizando a ferramenta *antispam*, e verificação de vírus. Executar o controle sobre as permissões de envio e recebimento de anexo por usuários e permissão de envio e recebimento de *email* para determinados endereços dos mesmos. Para controlar as políticas de envio e recebimento são atribuídos endereços e anexos, que o usuário não terá permissão às categorias definidas para envio ou para recepção, ou seja, o *email* será rejeitado. Esta rejeição se dará através de um *email* que o remetente receberá após efetuar o envio.

Dentre os procedimentos metodológicos empregados foi realizado o estudo inicial do problema e análise dos requisitos funcionais, em seguida, definida uma proposta de plataforma (modelagem do Ironmail). Feito isso, foi obtida a implementação de um protótipo da plataforma proposta (desenvolvimento da ferramenta), para que fosse desempenhada a avaliação e validação da proposta (testes, resultados e conclusões).

O Artigo está organizado da seguinte forma: na seção 2 é apresentado o que é um SMTP *Proxy* e as ferramentas utilizadas no Ironmail, na seção 3 serão apresentados os trabalhos relacionados. Na seção 4 é esplanada uma breve descrição sobre o problema proposto, bem como toda a modelagem da aplicação. Na seção 5 é mostrada a solução. Na seção 6 são apresentados os testes, com os respectivos resultados, executados para comprovar seu funcionamento. Na seção 7, são apresentadas as conclusões e as sugestões para trabalhos futuros.

2. Conceitos Básicos

Os *SMTP Proxies* [Lavigne 2003], têm o funcionamento semelhante ao *proxy* http, porém, para servidores de *email*. Surgiram para combater grande parte dos problemas causados por *emails* indesejados, bloqueando até 99% de ameaças reais. Um filtro de *emails* é executado no servidor de *emails* e trabalha como filtro dos pacotes SMTP - *simple mail transfer protocol* – [Klensin 2001], efetuando o controle das permissões de conteúdo dos usuários sobre seus *emails*.

2.1. Tecnologias Utilizadas

Para reduzir o tráfego nas redes e a perda de tempo e dinheiro com remoções de vírus e, até mesmo, para proteger os usuários de softwares maliciosos como os Cavalos de Tróia (*trojan horses*) foram desenvolvidas algumas ferramentas, *freeware* e comerciais. Dentre estas ferramentas, podem-se destacar algumas que constam neste projeto. São elas:

- Spamassassin [Spamassassin 2007]: Trata-se de um módulo para identificação de *spams* utilizando análise do texto, consultas em tempo real a listas negras, análises estatísticas e algoritmos de *hash*. Executa testes de heurística no cabeçalho e no corpo do *email* para identificar o *spam*. Uma vez identificado, é inserida uma marca em seu cabeçalho.

- ClamAV Antivírus [Kojm 2007]: O Clam Antivírus é um antivírus *open source* que possui versões para Windows (o ClamWin) e Linux, com atualizações diárias. É disponibilizado pela licença GPL - *General Public License* [Gnu 2007]. O ClamAV para Linux pode ser usado por linha de comando, ou efetuando o *download* de um pacote especial para ser usada uma interface gráfica pelo pacote ClamTK.

Para armazenamento dos dados é utilizado o SGDB MySQL [Sun Microsystems 2008], já o desenvolvimento da aplicação foi utilizada a linguagem de programação PERL [Christiansen; Torkington 2007] por sua facilidade em tratar arquivos de texto e criar *daemons*, ou serviços. A interface gráfica foi desenvolvida em pascal utilizando o Lazarus [Free Pascal 2008], ferramenta *open source* extremamente semelhante ao Delphi, seu compilador é o *Free Pascal Compiler*.

Como servidor de *emails* foi escolhido o Postfix [Venema 2007; Dent 2003] por ser um dos mais conhecidos, seguros, com grande quantidade de documentação e comunicação direta com o banco de dados MySQL. O que facilita a utilização de domínios virtuais, ou seja, a utilização de vários domínios para um mesmo endereço IP.

3. Trabalhos Relacionados

Alguns trabalhos foram cuidadosamente estudados e até suas atuais versões, Amavisd-new 2.52 e Mailscanner 4.65, respectivamente. Estas são as ferramentas mais utilizadas no mercado e possuíam as seguintes propriedades:

- **Amavisd-new** [Martinec 2007] - Efetua filtragem de *emails* por usuários, por endereços e anexos, *antispam* utilizado o Spamassassin e executa verificação de vírus. Sua desvantagem é que efetua apenas testes globais de permissões para envio de anexos. Ocorrem muitos falsos positivos, ou seja, *emails* bloqueados indevidamente.
- **Mailscanner** [Field 2007] - Efetua filtragem de *emails* por endereços e anexos, *antispam* utilizado o Spamassassin e o antivírus executa verificação de vírus. Sua desvantagem é que efetua apenas testes globais.

4. O que é o Ironmail

Os MTAs - *Mail Transfer Agent* – [Dent 2003], ou popularmente chamados de servidores de *email*, possuem em suas funções a filtragem ou bloqueio para todo o tipo de *email* indesejado. A opção é utilizar ferramentas externas para que efetuem as filtragens com permissões para envio e recepção do *email*.

O Ironmail é um gerenciador para políticas, sob licença GPL, determinadas pelo administrador da rede, de envio e recebimento de *emails*. Pode controlar as permissões de anexos e endereços. Efetua o bloqueio de *spams* utilizando *scores,* retornados pelo Spamassassin e bloqueio de anexos com vírus através do ClamAV Antivírus. Estes controles e políticas visam uma menor ociosidade dos usuários da rede decorrente do recebimento de *emails*. O Ironmail possui uma interface gráfica para gerenciar as políticas e um *daemon*, ou serviço, chamado ironmaild, para executar as políticas definidas sobre os *emails*.

4.1. Objetivos

Os principais objetivos da ferramenta são:

- Todos os *emails* tanto para o envio e recepção, são filtrados pelo Ironmail;
- Filtro *antispam* usando Spamassassin onde é gerado um valor chamado *score.* Este *score* advém de resultados de cálculos estatísticos efetuados pela ferramenta, quanto mais alto o *score*, maior a chance de ser um spam. O Ironmail, dependendo deste *score*, definirá se o email será simplesmente excluído, ou enviado para uma quarentena ou inserida uma *tag SPAM* no campo assunto do email;
- Filtragem de vírus utilizando o ClamAV Antivírus. A filtragem é efetuada pelo *daemon* do ClamAV, o clamd. Caso haja falha na conexão por *socket*, o ironmaild efetua a verificação em linha de comando. O ClamAV Antivírus foi definido como antivírus padrão por ser um antivírus *freeware*, porém é possível configurar junto ao ironmaild, algumas outras ferramentas de antivírus, inclusive comerciais dentre elas estão o F-Secure FSAV [F-Secure 2008], Trend Micro Vscan [Trend Micro 2008], BitDefender BDC [Bitdefender 2008], Symantec [Symantec 2008] e Kaspersky [Kapersky 2008], como os mais conhecidos.
- Interface gráfica para cadastro, edição e exclusão de usuários, endereços de *email*, categorias de endereços e anexos, com o intuito de facilitar o gerenciamento das políticas, definidas pelo administrador da rede, para os usuários;
- Os *emails* bloqueados podem ser excluídos permanentemente ou enviados para uma caixa postal de quarentena;
- É possível criar políticas de usuário por:
  a) Endereços de *emails* - o usuário terá permissão de envio ou recepção para um determinado endereço, previamente cadastrado pelo administrador da rede.
  b) Anexos - o usuário terá permissão de envio ou recepção para determinado anexo, previamente cadastrado pelo administrador de rede.
  c) Listagem com endereços de *email* e domínios para liberação ou bloqueio global (*White/Black list*).

4.2. Ironmaild

O ironmaild é o *daemon*, ou servidor, do Ironmail. Ele é o responsável por aplicar todas as políticas definidas pelo administrador em um serviço executado em *background.* Ao mesmo tempo em que é um cliente, ou seja, envia *emails* para outro servidor de *emails*, é servidor SMTP, pois permanece ouvindo requisições de envio de *emails* solicitadas pelo Postfix. Seu trabalho é receber o *email* do Postfix, executar seus procedimentos e devolver novamente ao Postfix para a entrega da mensagem.

O diagrama de classes do ironmaild (figura 1) possui poucas classes. Efetua, basicamente, tratamento de *strings* e executa comandos em arquivos. A classe principal é o próprio Ironmail. Ele irá gerar nosso processo principal com o Ironmail::Server, recupera os dados e políticas dos usuários, incluindo sua categoria. Após executar seus

testes retornará o *email* ao Postfix utilizando o Ironmail::Client, em um endereço e porta determinados.

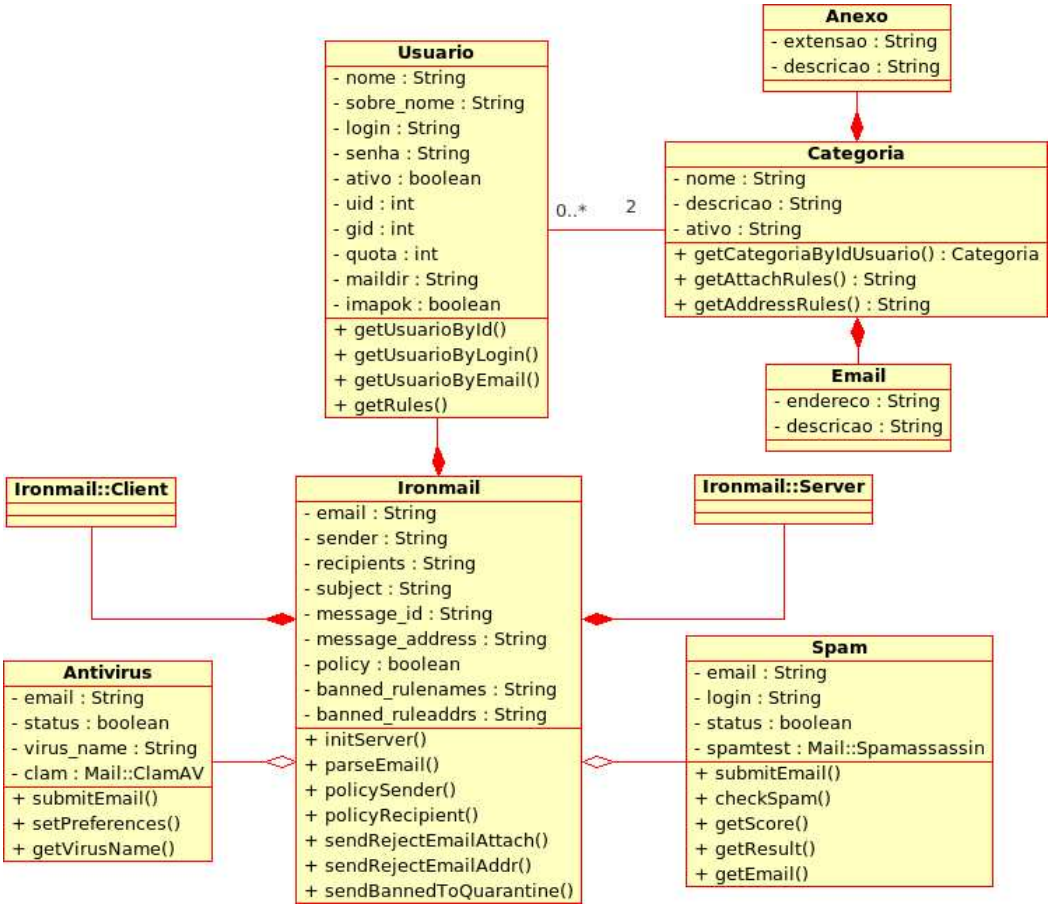

Figura 1. Diagrama de Classes

4.3. Requisitos para envio de *email*

Os requisitos para envio de *email* através do sistema são:

- Receber o *email* através do protocolo SMTP
- Verificar se o usuário tem permissão de envio para o destinatário, tanto para o endereço de *email* quanto para o domínio.
- Verificar se o *email* possui anexo.
- Se o *email* possuir anexo verificar o tipo de anexo por seu cabeçalho.
- Verificar se o usuário possui permissão de envio para o anexo.

4.4. Requisitos para recepção de *email*

Os requisitos para recepção de *email* através do sistema são:

- Receber o *email* através do protocolo SMTP
- Salvar o *email* em *Spool.*
- Verificar se o *email* é *spam* através do *Spamassassin*.

- Verificar se o usuário tem permissão de recebimento de *email* do remetente, tanto para o endereço de *email* quanto para o domínio.
- Verificar se o *email* possui anexo.
- Se o *email* possuir anexo verificar o tipo de anexo por seu cabeçalho.
- Verificar se o usuário possui permissão de envio para o anexo.
- Verificar a existência de vírus no anexo.

4.5. Diagrama ER

Neste tópico é apresentado o diagrama ER (figura 2) onde serão armazenados os dados. A principal entidade é o usuário. Ele poderá possuir mais de um *email*, ou seja, seu usuário terá relacionamento com mais de um domínio através da tabela de DNS. Possuirá ainda uma categoria. As categorias servirão como perfis e agregarão anexos, preferências de *spams*, tabela *spams*, e categoria de *emails* onde serão armazenados os *emails* de *whitelist* ou *blacklist* para os usuários. Por fim há uma entidade para *whitelist* e *blacklis*t globais que servirá para armazenar endereços que deverão ser liberados ou bloqueados independentemente das regras definidas aos usuários.

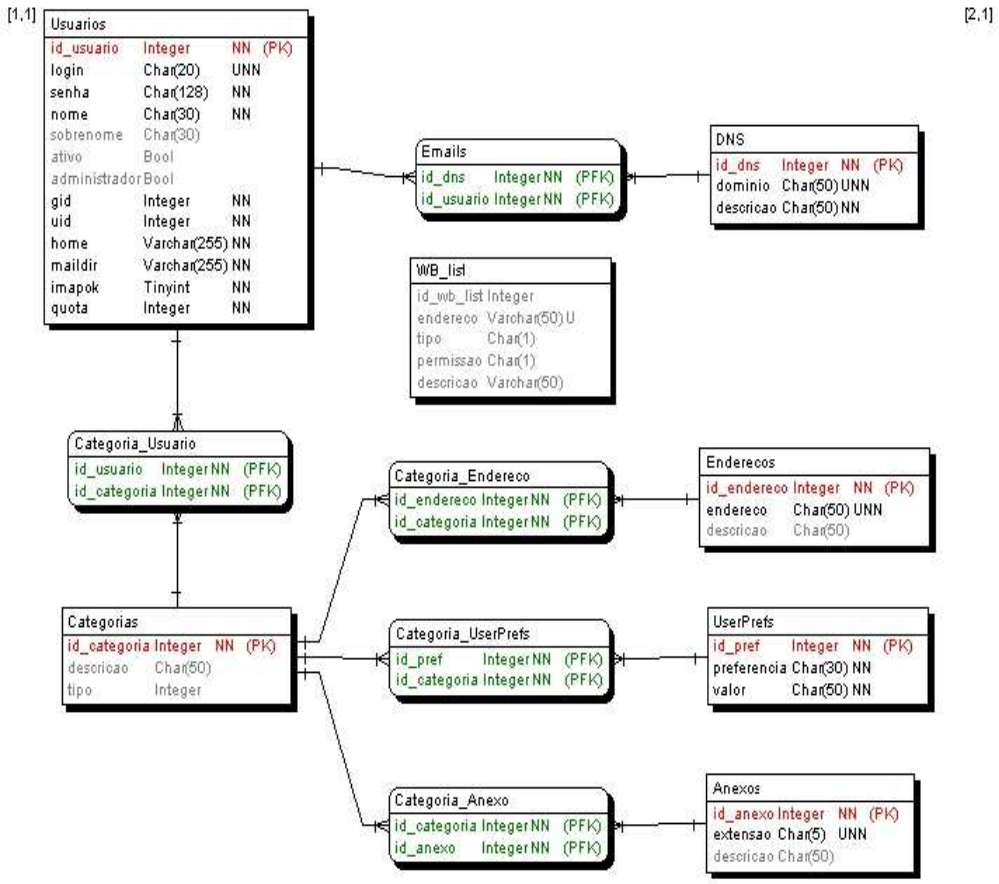

Figura 2. Diagrama ER

5. Fluxo de Trabalho

Como Ironmail não é apenas um *software* - é uma solução contra pragas oriundas de *emails* - para uma melhor compreensão do fluxo de trabalho, é preciso conhecer um pouco mais sobre o funcionamento da ferramenta.

Ele utiliza como servidor de *emails* um dos mais conhecidos no mundo *freeware/open source*, o Postfix. Este servirá de porta de entrada para o *proxy* desenvolvido.

O fluxo do *email* (conforme figuras 3 e 4, para envio e recepção, respectivamente) no servidor inicia quando um *email* chega à porta 25 (SMTP) (ilustrado pela figura 5). O servidor de *emails* utilizado (Postfix) possui algumas proteções extras como testes. Estes testes verificam falhas de configurações nos servidores de *email*, características comuns de *Spammers*. Dentre estes testes possuem destaque:

- Listas RBL [Lavigne 2003], *realtime blacklists* – listas negras, de endereços, endereços IP e domínios, consultadas em tempo real.
- Verificações de IP válido – verifica se o endereço IP realmente existe. Nesta verificação é necessária uma correta configuração do DNS (*Domain Name System*) reverso do servidor de resoluções de nomes de domínios.
- Verificações de DNS [Registro.br 2007] – verifica se o endereço de DNS corresponde ao endereço IP do *mail server* do remetente.

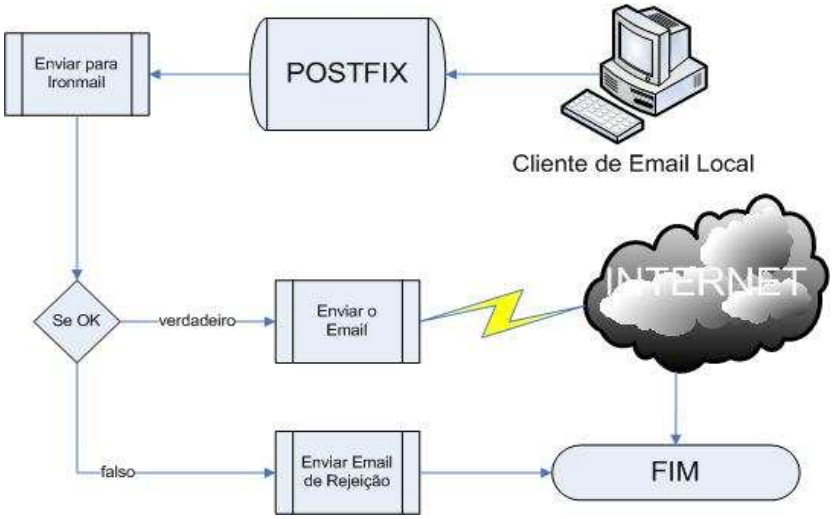

Figura 3. Fluxo de envio do email

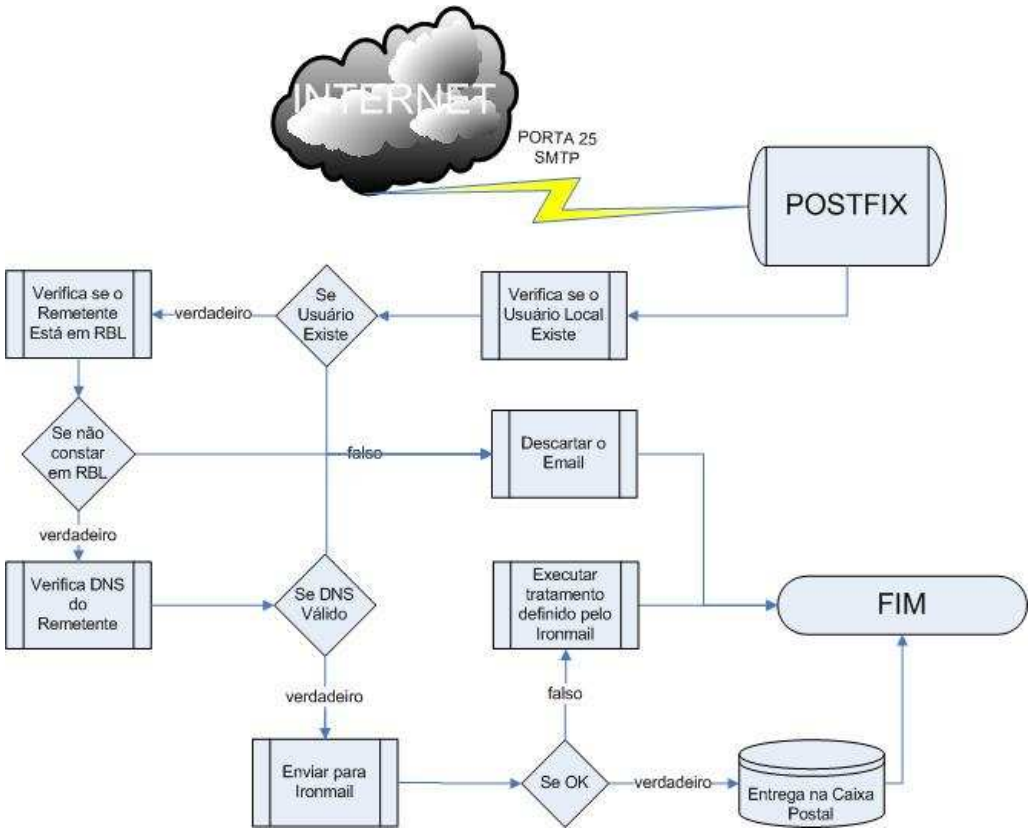

Figura 4. Fluxo de recepção do email

Após receber o *email,* o Postfix, ouvindo na porta 25, SMTP padrão, efetua seus testes. Efetuados os testes, é realizado um redirecionamento da mensagem para o ironmail que estará escutando na porta 10024, definida como padrão.

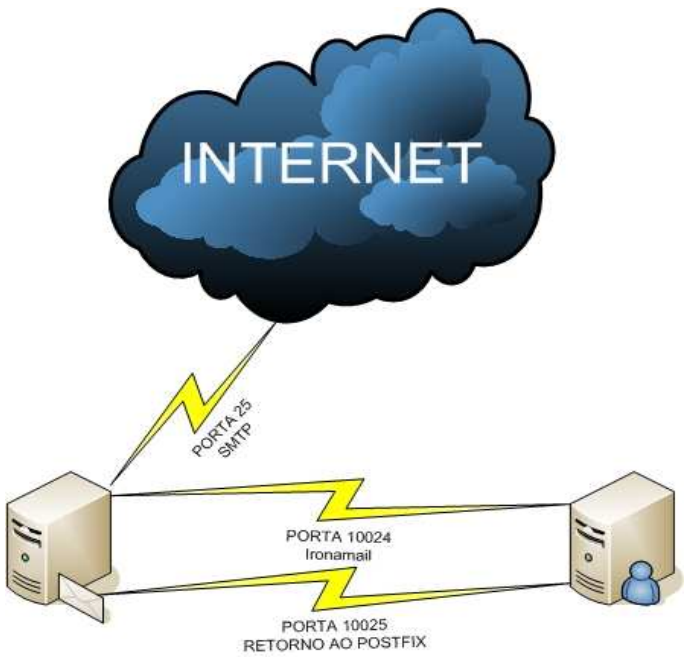

Figura 5. Tráfego sobre o protocolo

Com estas verificações extras executadas diretamente no servidor de *emails,* pode-se poupar processamento e memória, pois os *emails*, se não passarem nestes testes, serão descartados antes mesmo de serem enviados para a filtragem, motivo pelo descarte destas implementações.

Todos os acontecimentos, dentro do servidor, possuem códigos de *status* que informam o estado atual da mensagem, os principais códigos (utilizados pelo ironmail) são apresentados de acordo com a tabela 1. No caso de um *email* originado de um servidor de *emails* com DNS mal configurado, que é caso de servidores *spammers*, é retornado o código de status 521, por exemplo. Se entregue com sucesso, na caixa postal do usuário, é retornado o código 250. Quando estiver esperando para entrega, por algum motivo, como caixa postal indisponível, o código retornado é 450.

Tabela 1. Códigos de status retornados por transações SMTP

Código	Descrição
250	OK, queuing for node node started. Requested mail action okay, completed.
450	Requested mail action not taken: mailbox unavailable. ATRN request refused.
452	Requested action not taken: insufficient system storage.
500	Command not recognized: command. Syntax error.
550	Requested action not taken: mailbox unavailable.
552	Requested mail action aborted: exceeded storage allocation.
553	Requested action not taken: mailbox name not allowed.
554	Transaction failed.

6. Testes Executados

Para comprovar que o Ironmail está executando suas funcionalidades definidas, foram executados alguns testes funcionais. Os testes executados para o envio de *emails* foram os seguintes:

- Envio de 100 *emails* classificados como *spam* por outros filtros de *email* contra o servidor de *emails* configurado para utilizar o Ironmail como filtro.

  Resultado: Dos 100 *emails* enviados contra o usuário local, 7% chegaram ao destinatário. Isto acontece, pois os *emails* foram extraídos de servidores em atividade, conseqüentemente, com um ensinamento maior do Spamassassin, pois ele possui um *autolearn* ou auto-aprendizado.

- Envio de *email* com arquivo Eicar [Eicar 2008] anexo contra o servidor de *emails* configurado para utilizar o Ironmail como filtro. O Eicar foi criado para efetuar testes em ferramentas de antivírus.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados com o arquivo Eicar foram bloqueados.

- Envio de *email*, para endereço externo, não permitido para o usuário local utilizado.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados rejeitados.

- Envio de *email*, para endereço externo, com anexo com extensão não permitida para envio do usuário local utilizado.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados rejeitados.

- Envio de *email* para endereço externo bloqueado, de usuário local cadastrado.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados rejeitados.

- Envio de *email* para endereço externo bloqueado, com anexo e com extensão permitida, de usuário local cadastrado.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados rejeitados.

A figura 6 ilustra a compilação dos resultados obtidos nos testes de envio.

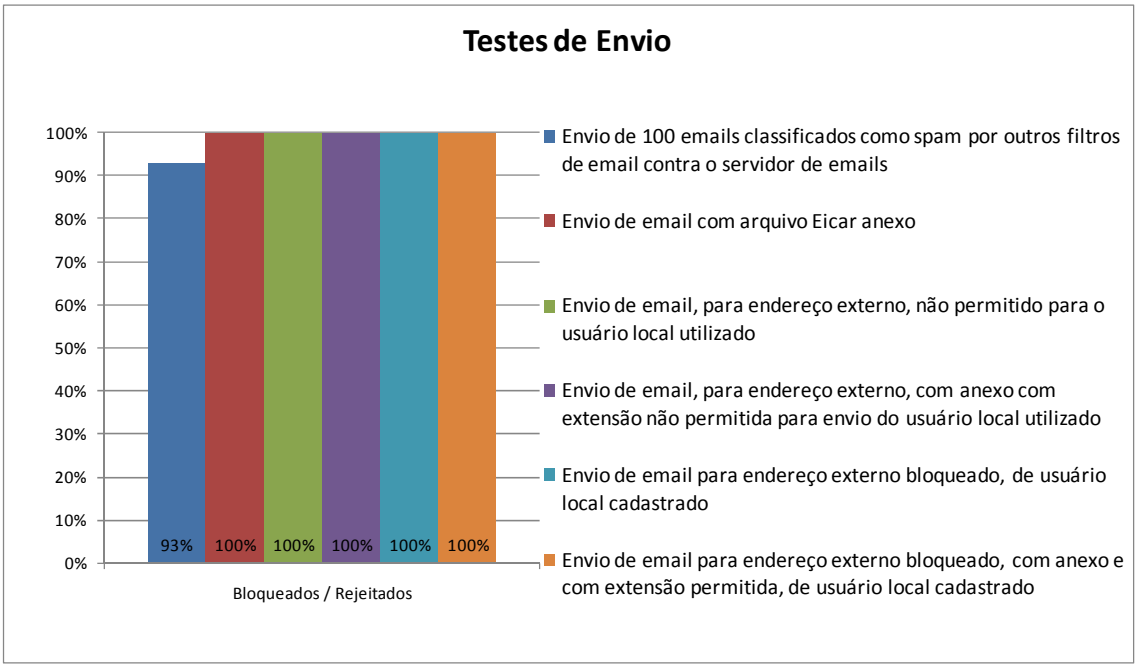

Figura 6. Testes de Envio de *email*

Em relação à recepção de *email* os seguintes testes foram realizados:

- Recepção de email, de endereço externo, não permitido para o usuário local utilizado.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados rejeitados.

- Recepção de email, de endereço externo, com anexo com extensão não permitida para recepção do usuário local utilizado.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados rejeitados.

- Recepção de email de endereço externo bloqueado, para o usuário local cadastrado.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados rejeitados.

- Recepção de email de endereço externo bloqueado, com anexo e com extensão permitida para recepção do usuário local cadastrado.

  Resultado: O resultado foi satisfatório, 100% dos *emails* enviados rejeitados.

A figura 7 ilustra a compilação dos resultados obtidos nos testes de recepção.

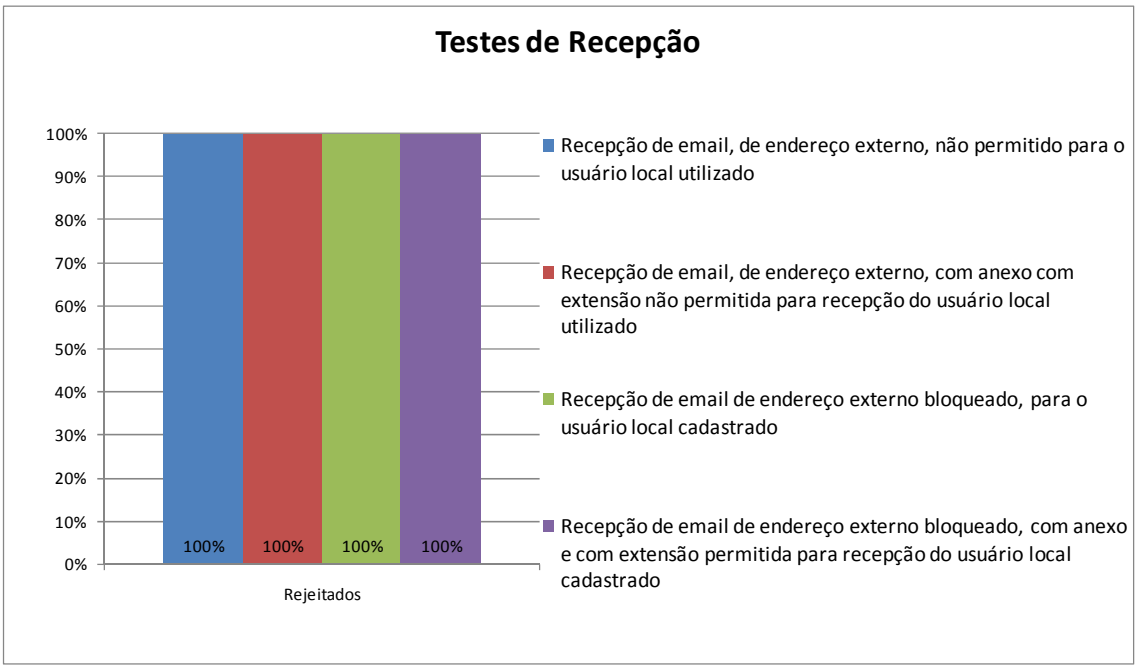

Figura 7. Testes de Recepção de *email*

7. Conclusões

Este trabalho permitiu constatar vários dos problemas complexos enfrentados pelos gestores da área tecnológica e, em particular, os problemas associados ao envio e recepção de *emails*. Os problemas relacionados ao envio e recebimento de *emails* podem ser solucionados a partir da utilização de políticas adequadas e bem definidas sobre os usuários do domínio.

Pode-se concluir que, com algumas restrições sobre o envio ou recebimento de *emails* os usuários passam a ter um foco maior em suas principais atividades. Um detalhe importante é que mesmo com a filtragem de *emails* indesejados os administradores de rede deverão efetuar verificações regulares na caixa postal de quarentena a fim de verificar falsos positivos que possam ocorrer e assim encaminhar o *email* a seu destinatário.

Os *spams* deixam de ser um problema quando 99% deles são bloqueados ainda mesmo no servidor de *emails*, inclusive *emails* que possuam textos inseridos em imagens. Passa a ser desnecessária a preocupação do usuário em filtrar *emails* que sejam *spams* em suas caixas postais.

Email com vírus, em anexo, passam a ser inexistentes se a ferramenta de antivírus estiver permanentemente com seu banco de dados atualizado. Finalmente, foi concluído que a utilização do Ironmail, ou outra ferramenta de filtragem de *emails*, é indispensável em qualquer servidor de *emails*.

7.1. Trabalhos futuros

Além das funcionalidades e características da aplicação apresentadas nesse trabalho os seguintes trabalhos futuros fariam parte do escopo deste trabalho:

- Armazenamento de dados sobre os *emails*, como: remetente, destinatários, assunto e tamanho da mensagem, para geração de relatórios e gráficos sobre os *emails* enviados e recebidos;
- Reescrever o ironmaild utilizando a linguagem de programação Java [Deitel e Deitel 2006] com o intuito de ter um melhor desempenho da aplicação;
- Autenticação de usuários por catálogo LDAP;
- Possibilidade de utilizar caixas postais de quarentena por usuário;
- Possibilidade de utilizar outros sistemas gerenciadores de banco de dados além do MySQL;
- Desenvolvimento de interface *web* para administração remota;
- Manipulação de regras para o Spamassassin para controle sobre palavras. Pode-se manipular o *score* de acordo com palavras encontradas no *email*;
- Remodelagem do algoritmo para facilitar a utilização de *plugins*.

Referências

Antispam.br. (2007). *O que é spam*?. Disponível em: www.antispam.br/conceito/

Bitdefender. (2008). *Antivirus Software – BitDefender*. Disponível em: http://www.bitdefender.com/

Christiansen, Tom; Torkington, Nathan. (2007). *Perldoc*. Disponível em: http://perldoc.perl.org

Deitel, Harvey; M., Deitel, Paul J. (2006). *JAVA Como Programar*. 6ª edição. São Paulo: Editora Pearson Prentice Hall, 2005.

Dent, Kyle D. (2003) *Postfix: The Definitive Guide*. 1ª edição. Santa Clara, CA: Editora O'Reilly, 2003.

Eicar. (2008). *The anti-virus or anti-malware test file*. Disponível em: http://www.eicar.org/anti_virus_test_file.htm

Field, Julian. (2006). *Mailscanner*. Mailscanner Documentation. Disponível em: http://www.mailscanner.info

Free Pascal. (2008). *Lazarus*. Free Pascal Lazarus Project. Disponível em: http://www.lazarus.freepascal.org/

F-Secure. (2008). *Antivirus and Intrusion Prevention Solutions for Home Users and Business*. Disponível em: http://www.f-secure.com/

Gnu. (2007). *General Public License*. Disponível em: www.gnu.org/copyleft/gpl.html

Kapersky. (2008). *Kapersky Lab*. Disponível em: http://www.kaspersky.pt/

Klensin, J. (2001). *Simple Mail Transfer Protocol*. Disponível em: http://www.ietf.org/rfc/rfc2821.txt

Kojm, Tomasz. (2007). *Clamav*. Clamav Documentation. Disponível em: http://www.clamav.org/

Lavigne, Dru. (2003). *SMTP Proxies*. Disponível em: http://www.onlamp.com/pub/a/bsd/2003/07/24/FreeBSD_Basics.html.

Martinec, Mark. (2007). *Amavisd-new*. Amavisd-new Documentation. Disponível em: http://www.ijs.si/software/amavisd/

Michellis, Deives. (2004). *Técnicas Anti-SPAM com o Postfix - Parte 1*. Disponível em: http://www.unitednerds.org/thefallen/docs/index.php?area=Postfix&tuto=Postfix-AntiSpam-1

Registro.br. (2007). *FAQ (Perguntas Freqüentes)*. Disponível em: http://registro.br/faq/faq5.html

Resnick, P. (2001). *Internet Message Format*. Disponível em: http://www.ietf.org/rfc/rfc2822.txt

Spamassassin. (2007). *The Apache Foundation*. Disponível em: http://spamassassin.apache.org/

Sun Microsystems. (2008). *MySQL*. Disponível em: http://www.mysql.com/

Symantec. (2008). *Symantec Corporation.* Disponível em: http://www.symantec.com/

Todd, Bennett. (2007). *SmtpProx*. Smtpprox Documentation. Disponível em: http://bent.latency.net/smtpprox/

Trend Micro. (2008). *Antivirus and Content Security Software.* Disponível em: http://us.trendmicro.com/

Venema, Wietse Z. (2007). Postifix. Disponível em: http://www.postfix.org