TESTES DE FERRAMENTAS OPEN SOURCE NO COMBATE AO SPAM

TESTING ANTI-SPAM OPEN SOURCE TOOLS

André Gonzatti, Marco Antônio Trentin, Adriano Canabarro Teixeira

Curso de Ciência da Computação, Universidade de Passo Fundo, Bairro São José, BR- 285,
CEP: 99052-900 - Cx. Postal 611.E-mail: andregonzatti@upf.br, trentin@upf.br,teixeira@upf.br

RESUMO

O grande número de mensagens indesejadas recebidas pelos usuários em suas caixas de correio eletrônico faz com que se busquem formas de combater ou amenizar os problemas ocasionados pelo recebimento de spams. O presente trabalho destina-se ao estudo de técnicas e ferramentas que podem ser utilizadas no combate ao spam, apresentando o funcionamento do correio eletrônico e do protocolo SMTP, as formas mais comuns de spams existentes atualmente, estatísticas de ocorrência, bem como as formas utilizadas para envio desse tipo de mensagem. Também será mostrado o funcionamento e um comparativo dos testes realizados com três ferramentas open source: Bogofilter, SpamAssassin e SpamPal destinadas ao combate ao spam.
Palavras-chave: spam, correio eletrônico, SMTP

ABSTRACT

The large number of unwanted messages received by users in their mailboxes, makes them look for ways to combat or alleviate the problems caused by the receipt of spam. The present work aims to study the techniques and tools that can be used to combat spam, showing the operation of electronic mail and the SMTP protocol, the most common forms of spam currently existing statistics of the occurrence, as well as the forms used for sending this message type. Will also be shown the operation and a comparison of three tests with open source tools: Bogofilter, SpamAssassin and SpamPal designed to combat spam.
Keywords: spam, eletronic email, SMTP

1. INTRODUÇÃO

Com a expansão da internet, o e-mail se tornou uma das formas mais utilizadas para comunicação entre pessoas, facilitando e agilizando a comunicação entre elas. Com toda essa facilidade também surgiram alguns problemas para os usuários, como o recebimento excessivo de mensagens não solicitadas, o que denominamos *spam*.

Dentre os malefícios trazidos pela prática de envio de *spam*, tem-se o consumo de banda de internet, aumento da carga em servidores e clientes de e-mail, sem contar o uso desses para prática de golpes, visando obter lucros de forma ilícita, com o acesso a informações confidenciais dos usuários. Observando esse cenário, fica clara a importância de se conhecer meios para diminuir ao máximo o impacto causado por esse tipo de ataque (KUROSE, 2009).

O objetivo desse trabalho é conhecer mais detalhadamente o funcionamento do correio eletrônico bem como o protocolo SMTP, apresentando o cenário do *spam* no Brasil e no mundo, destacando os principais tipos de *spams* e as formas de ataque que são mais utilizadas pelos *spammers*.

Em seguida serão abordadas as principais técnicas que podem ser utilizadas no combate ao *spam* e o estudo detalhado de três ferramentas de código aberto, mostrando o seu funcionamento e apresentando a sua eficácia diante de diferentes configurações.

Com uma base de e-mails de seis contas distintas, são demonstrados testes de filtragem de *spam* com as ferramentas *Bogofilter*, *SpamAssassin* e *SpamPal* e os resultados são comparados para verificação de eficácia de cada uma delas.

Ao final serão apresentadas as conclusões pertinentes sobre esse estudo e da utilização de cada uma das ferramentas propostas.

2. FERRAMENTAS ANTI-SPAM OPEN SOURCE

Nesse capítulo serão apresentadas as três ferramentas *open source* utilizadas nesse trabalho. As mesmas foram as escolhidas pois são as mais utilizadas atualmente nas organizações que fazem uso de servidores de email.

2.1 Bogofilter

O Bogofilter é um filtro utilizado no MTA - *Mail Transfer Agent*, que realiza uma análise estatística do cabeçalho e do conteúdo do e-mail, classificando-o como spam ou não, sendo possível treiná-lo, incluindo classificações e correções do usuário.

A técnica estatística de classificação é conhecida como técnica Bayesiana e seu uso para spam foi descrito por Graham (2002) em seu artigo “Um Plano Para Spam”. Gary Robinson, em seu weblog Rants, sugeriu alguns aperfeiçoamentos para melhorar a discriminação entre spam e não spam.

A versão inicial foi escrita por Eric S. Raymond em linguagem C, em 2002. Atualmente é desenvolvido por David Relson, Andree Matthias, Louis Greg e um grupo de voluntários e é disponibilizado sob licença GPL (BOGOFILTER, 2011).

A origem do nome Bogofilter vem da palavra *bogus* que significa incorreto ou não funcional, e ele funciona tratando cada entrada (e-mail) separando-os em *tokens* e então cada um deles é verificado, comparando-se a uma lista de palavras que fica armazenada no banco de dados Berkeley DB. Nesse banco de dados fica guardada uma contagem do número de vezes que certo *token* ocorreu em determinado e-mail considerado *spam* e não *spam*. Aplicando-se a teoria da probabilidade bayesiana é possível então estimar através desses números a probabilidade de certa mensagem ser ou não *spam*.

Através de cálculo de estimativa de *tokens* é encontrado um valor que indica o quanto não é *spam* o e-mail analisado e outro cálculo é feito para indicar o quanto *spam* o mesmo é considerado. Da subtração desses dois valores obtêm-se um indicador combinado denominado *bogosity*, que fica próximo de 0 se os indícios de que não é *spam* são fortes e próximos de 1 se provavelmente é um *spam*. Se os dois sinais são iguais o valor ficará próximo de 0,5, como pode ser visto na figura a seguir:

```
H 0.000070
S 0.993996
U 0.505995
```

Figura 1 - Indicador *bogosity*: H (ham), S (spam) ou U (unsure)

O Bogofilter pode trabalhar no modo três estados, quando a "bogosidade" (*bogosity*) tiver valor próximo de 0 a mensagem é marcada como *ham*[1] (ou seja, não é spam), quando tiver valor próximo de 1, a mensagem é marcada como *spam* ou se estiver próximo de 0,5 ela é marcada como *unsure* (incerta). No modo dois estados não é considerado o modo *unsure*, somente *ham* ou *spam*.

Há parâmetros que influenciam nesse cálculo, podendo ser citado o *spam_cutoff*, onde as mensagens com "bogosidade" igual ou maior que esse valor são marcadas como *spam* e o *ham_cutoff*, quando esse valor é menor ou igual a *ham_cutoff* as mensagens serão marcadas como não *spam*. No modo 3 estados os valores entre *ham_cutoff* e *spam-cutoff* são marcados como incertos (ZUCCO, 2005).

Há quatro formas de realizar o treinamento do Bogofilter, e quanto maior o número de mensagens utilizadas para seu treinamento melhor será a exatidão na classificação das mensagens. Dentre essas formas podemos citar:

- **Método 1 – Treinamento completo:** cria uma base de palavras no banco de dados pontuando cada uma delas de acordo com o número de vezes que aparece na base de treinamento. A seguir segue exemplo de treinamento com esse método:

```
bogofilter -s  < spam.mbox
bogofilter -n  < ham.mbox
```

Figura 2 - Exemplo de treinamento completo do *Bogofilter*

Na primeira linha é feito o treinamento indicando ao Bogofilter quais mensagens são consideradas *spam*, enquanto que na segunda, quais não são. Cada caixa de mensagens está no formato *mbox*.

- **Método 2 – Usando o script bogominitrain.pl**: esse *script* verifica as mensagens dentro da caixa de cada usuário e quando o *script* confirma que a base de dados classifica as mensagens corretamente ele pára de executar. Para isso o usuário deve ter classificado em sua caixa postal o que é e o que não é *spam* e separado em pastas. A base de dados é atualizada até que for constatado que a classificação das mensagens é feita corretamente. Pode ser usado a opção *-o* do *script*, definindo assim uma margem de erro aceitável sobre a variável *spam_cutoff*.

```
bogoninitrain.pl -FNV  /home/user/.bogofilter ham.mbox spam.mbox '-o 0.9,0.3'
```

Figura 3 - Exemplo de treinamento utilizando o bogominitrain.pl

- **Método 3 – Usando o script randomtrain**: é gerada uma lista de todas as mensagens nas caixas de correio dos usuários e analisa aleatoriamente todas as mensagens, colocando um *score* para cada uma até que classifique corretamente assim como no método 2. A pré-classificação das mensagens pelo usuário também se faz necessária nesse método e ele se mostra mais eficaz utilizando-se um grande número de mensagens.

```
rondomtrain -s spam.box -n ham.box
```

Figura 4 - Exemplo de treinamento utilizando o rondomtrain

- **Método 4 – Train-on-error:** é realizado o treinamento com o método 1. Depois é refeito o treinamento com as mensagens que foram classificadas como *unsure* ou incorretamente. Há dois *scripts* disponíveis no *Bogofilter* que utilizam a técnica *train-on-error*. (BOGOFILTER, 2011).

[1] *ham* significa presunto, em inglês

No diretório */etc* do Sistema Operacional Linux fica localizado um arquivo denominado "bogofilter.cf", o qual possui as configurações do *Bogofilter*.

2.2 SpamAssassin

O *SpamAssassin* é um filtro *anti-spam* que realiza a filtragem das mensagens utilizando vários mecanismos incluindo a análise de texto, filtragem Bayesiana, listas de bloqueio DNS, e bases de dados de filtragem colaborativa. Desenvolvido em *Perl*, o *SpamAssassin* é um projeto da *Apache Software Foundation* (ASF) e distribuído sob licença GPL. Ele pode ser usando tanto em servidor como em cliente de e-mail e também é compatível com vários sistemas, como procmail, sendmail, Postfix, qmail, entre outros.

O *SpamAssassin* utiliza uma ampla gama de testes heurísticos, em cabeçalhos e no texto contido no corpo do e-mail para identificar *spams*. Dentre essas técnicas utilizadas podem ser citadas (SPAMASSASSIN, 2011):

- **Análise de Cabeçalho:** os *spammers* usam várias técnicas para mascarar suas identidades e esconder o servidor de origem das mensagens. O *SpamAssassin* tenta identificar indícios do uso desses "truques";
- **Análise de Texto:** o *spam* tem um estilo de texto próprio, geralmente destinado a lhe convencer de que um determinado produto ou serviço anunciado é uma oportunidade única e que não deve ser desperdiçada, além de tentar lhe convencer de que está recebendo essa mensagem porque se cadastrou em algum serviço ou porque um "amigo" o indicou. O *SpamAssassin* tenta identificar tal estilo, baseado em ocorrências comuns de palavras, frases, texto em MAIÚSCULAS ou "E N T R E S P A Ç A D O", entre outros;
- **Listas Negras:** o *SpamAssassin* suporta consulta a listas negras como mailabuse.org e ordb.org, e pode ignorar mensagens vindas de domínios reconhecidamente abusados por *spammers*;
- **Aprendizado na classificação:** o *SpamAssassin* possui módulos para a utilização de filtros Bayesianos, ou seja, na classificação das mensagens faz uso de estatística e probabilidades. Para funcionar corretamente é necessário treiná-lo;
- **Razor:** o *Vipul's Razor* é uma base de dados colaborativa para rastreamento de *spam*. Ela permite que um usuário reporte uma mensagem como *spam*, adicionando-a a base de dados do projeto, o que fará com que, automaticamente, todos dos outros usuários do *Razor* passem a ignorar a mensagem.

Há uma centena de testes[2] que são executados e para cada um deles em que a mensagem for reprovada, é atribuída uma nota. Se no final a mensagem tiver uma nota acima de 5 (pontuação padrão estipulada) a mensagem é classificada como *spam* (FABRE, 2011).

Para configuração do *SpamAssassin* há um arquivo chamado "local.cf" no diretório /etc/spamassassin. No endereço http://www.yrex.com/spam/spamconfig.php está disponível uma ferramenta para configurar e gerar esse arquivo de forma simples selecionando as opções que se quer.

Tendo as mensagens separadas em *spam* e *ham*, o treinamento do *SpamAssassin* pode ser feito como demonstrado na figura a seguir:

```
sa-learn --spam --mbox -C /etc/spamassassin Spam.mbox
sa-learn --ham –mbox -C /etc/spamassassin Ham.mbox
```

Figura 5 - Exemplo de treinamento do *SpamAssassin*

[2] http://spamassassin.apache.org/tests_3_3_x.html

No cliente de e-mail Mozilla Thunderbird ele já vem instalado como complemento, basta ativá-lo e ir treinando com as mensagens que são recebidas.

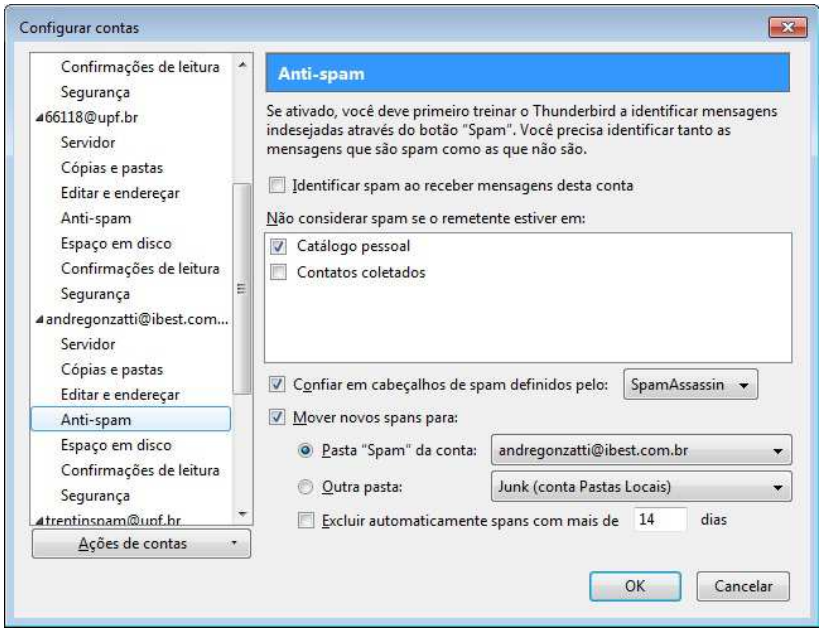

Figura 6 - Configuração do *SpamAssassin* no Thunderbird

Clicando no menu “ferramentas” e em “configurações da conta”, basta clicar na aba “Anti-spam”, marcar o item “Confiar em cabeçalhos de spam definidos pelo:” e selecionar o *SpamAssassin*. Também é possível mover as mensagens marcadas como *spam* para outra pasta e apagá-las automaticamente depois de um período pré-determinado.

2.3 SpamPal

O *SpamPal* é um filtro *anti-spam* de licença GPL, originalmente escrito e mantido por uma equipe de voluntários, que funciona como um servidor de e-mail *proxy* POP3/IMAP agindo entre o cliente de e-mail e o servidor de envio.

Funciona com vários programas de e-mail, como Outlook, Outlook Express, Eudora e Thunderbird, no qual também vem instalado como complemento, assim como o *SpamAssassin*. As mensagens são analisadas e marcadas com um cabeçalho “X-SpamPal: SPAM”, também ao assunto da mensagem é adicionado “** SPAM **” permitindo o uso de filtros para separar as mensagens consideradas *spam*.

Como principal meio utilizado para filtrar as mensagens o *SpamPal* compara os e-mails a *blacklists* (DNSBL), tais como SpamCop, além do treinamento com as mensagens recebidas no cliente de e-mail.

É possível adicionar *plugins* ao *SpamPal*, como filtros Bayesianos ou filtros de expressão regular (RegEx). Com o uso do *plugin* HTMLModify, por exemplo, é possível remover anexos perigosos das mensagens (como .bat, .src e .pif) os quais geralmente indicam um vírus. (SPAMPAL, 2011).

3. TESTE E ANÁLISE DAS FERRAMENTAS ANTI-SPAM OPEN SOURCE

Neste capítulo serão descritos o testes com as ferramentas *Bogofilter*, *SpamAssassin* e *SpamPal*, a fim de verificar a eficácia na classificação das mensagens, comparando qual delas em sua configuração *default* tem melhor resultado, relatando a incidência de falsos-positivos e falsos-negativos, bem como o nível de acerto de cada uma delas.

3.1 Ambiente de Testes

Para a realização dos testes com as ferramentas foi utilizado um notebook com processador AMD Turion 64 Dual - Core de 1.6 GHz, 2 GB de memória RAM e 120 GB de HD com sistema operacional Windows 7. Foi instalado a ferramenta VirtualBox da Oracle versão 4.1.4r74291 para virtualização do sistema operacional Linux Ubuntu 11.04. Também foi utilizado o cliente de e-mail Mozilla Thunderbird 8.0 para receber e armazenar as mensagens de e-mail utilizadas durante os testes e instalação do *Bogofilter*.

3.2 Base de Dados para Realização dos Testes

Como base de dados foi utilizada 8473 mensagens de seis contas distintas, sendo três delas pessoais, duas de outros colaboradores e uma de conta criada para receber as mensagens externas desviadas da conta de um usuário, sem passar por verificação de filtros *anti-spam* do servidor. Foram utilizadas mensagens do período de 25 de abril de 2005 a 31 de outubro de 2011.

3.3 Descrição dos Testes

As mensagens de cada conta de e-mail foram baixadas para o cliente de e-mail Mozilla Thunderbird em ambiente Windows, após foram copiadas para uma pasta denominada “Todas” e ordenadas por data. Dessa pasta foram selecionadas aleatoriamente 1000 mensagens e copiadas para duas pastas distintas, uma denominada “Spam” com 500 mensagens consideradas spams e outra “Ham” com 500 não spams as quais foram utilizadas para treinamento das ferramentas.

O arquivo que contém as mensagens no Thunderbird é do formato mbox, o qual pode ser lido pelas ferramentas sem precisar de conversão. Esses três arquivos (Todas.mbox, Ham.mbox e Spam.mbox) foram copiados para a máquina virtual para posterior realização dos testes.

Para o treinamento do *Bogofilter* foi utilizado o método 1 da seguinte forma:

```
bogofilter -s  -M < Spam.mbox
bogofilter -n  -M < Ham.mbox
```

Figura 7 - Comandos de treinamento do *Bogofilter*

Após o treinamento com as mensagens, foi realizado o teste do *Bogofilter* na base de dados que continha todas as mensagens recebidas e o resultado do teste gravado em um arquivo denominado “resultado_bogofilter” como descrito a seguir:

```
bogofilter -t -M  < Todas.mbox > resultado_bogofilter
```

Figura 8 - Comando de teste com a base de mensagens

No arquivo gerado cada linha indica uma mensagem a qual foi marcada com H (*ham*), S (*spam*) ou U (*unsure*) e também contém o índice de "bogosidade" de cada uma delas. Nesse arquivo foi ainda inserido um índice para posterior comparação com as mensagens analisadas de acordo com a figura a seguir:

```
H 0.000070 1
H 0.000000 2
S 1.000000 3
S 0.993996 4
H 0.000000 5
S 1.000000 6
H 0.036517 7
```

Figura 9 - Arquivo gerado com o teste do *Bogofilter*

Para posterior análise da classificação das mensagens feita pelo *Bogofilter*, foram selecionas de todas as mensagens, uma a cada intervalo de 29 e copiadas para outra pasta totalizando 292 mensagens, para então ser realizado o comparativo com o arquivo de teste obtido.

O treinamento do *SpamAssassin* e do *SpamPal* foi realizado através do Mozilla Thunderbird, selecionando as mensagens em cada pasta e com o botão direito marcando cada uma das pastas como *spam* ou não *spam* como demonstra a figura a seguir:

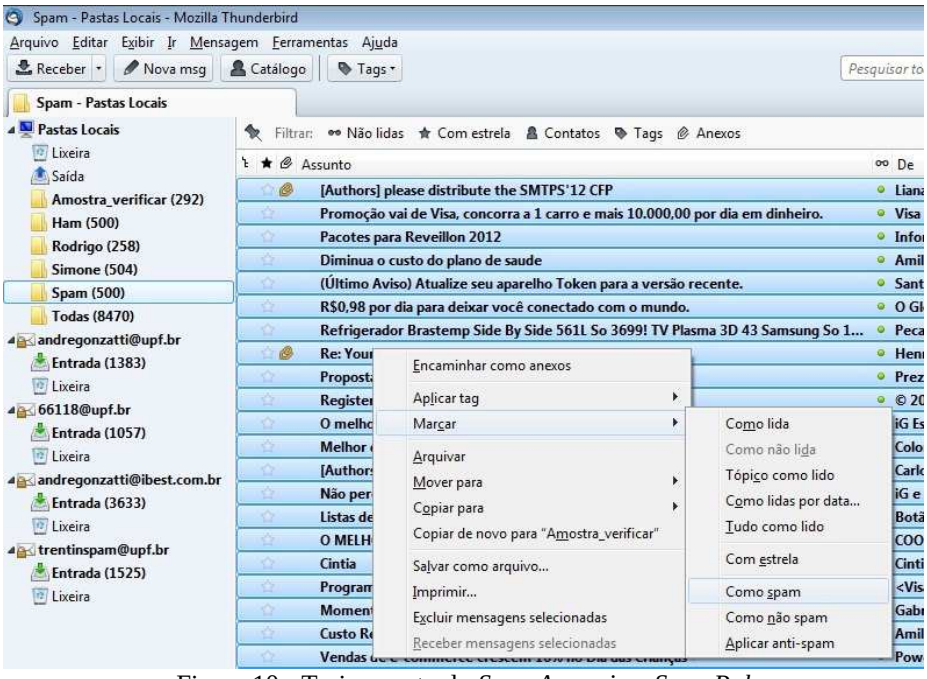

Figura 10 - Treinamento do *SpamAssassin* e *SpamPal*

Posteriormente para cada um deles foi aplicado o filtro *anti-spam* na pasta que continha a amostra das mensagens, as mesmas utilizadas para verificação de erros e acertos do Bogofilter. Após a aplicação do filtro cada mensagem considerada spam foi marcada com um símbolo, como uma pequena chama, em laranja.

3.4 Resultado dos Testes

O resultado dos testes será apresentado em tabelas e gráficos para melhor entendimento dos dados obtidos, bem como visualização do comparativo entre as ferramentas testadas. Para obtenção dos mesmos, foi realizada uma análise manual das mensagens selecionadas por amostragem (292 no total), comparando cada uma delas com a classificação realizada por cada uma das ferramentas analisadas.

Na tabela a seguir é possível verificar o comparativo entre as ferramentas após análise da amostragem comparando-se com o resultado dos testes.

Tabela 1 - Comparativo de resultados dos testes

Ferramenta	Acertos Spam	Acertos Ham	Falsos Positivos	Falsos Negativos
Bogofilter	42,27%	64,10%	13,70%	14,38%
SpamAssassin	89,69%	98,92%	4,45%	3,08%
SpamPal	93,10%	93,84%	4,45%	5,14%

Nos gráficos que seguem é possível comparar o índice de acertos na detecção de *spams* e não *spams*, bem como a ocorrência de falsos positivos e negativos de cada uma das ferramentas testadas.

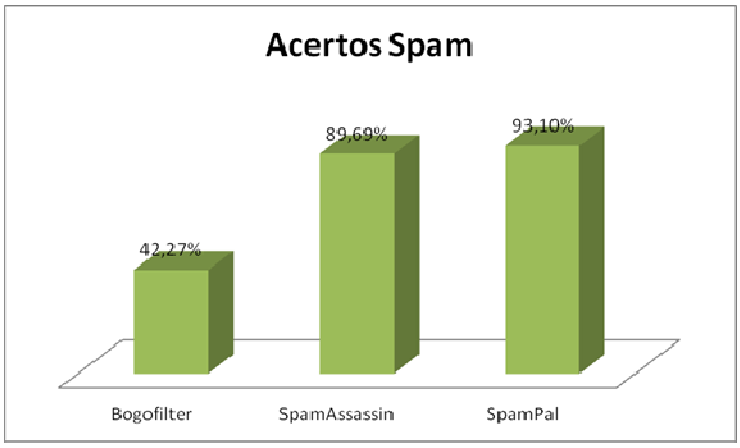

Figura 11 - Gráfico comparativo de detecção de *spam*

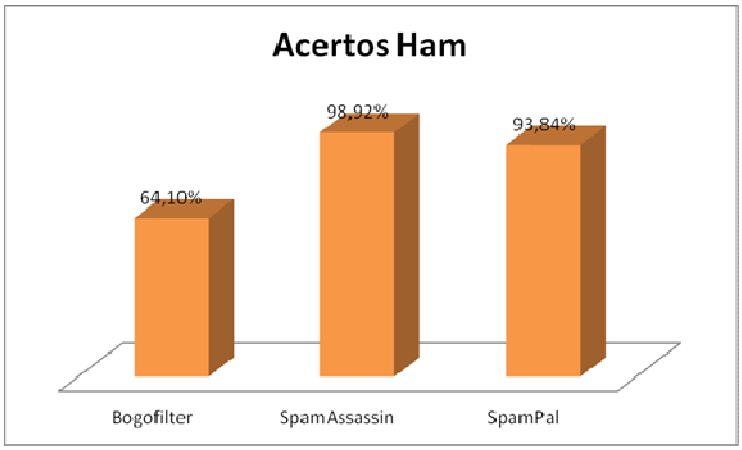

Figura 12 - Gráfico comparativo de detecção de não spam

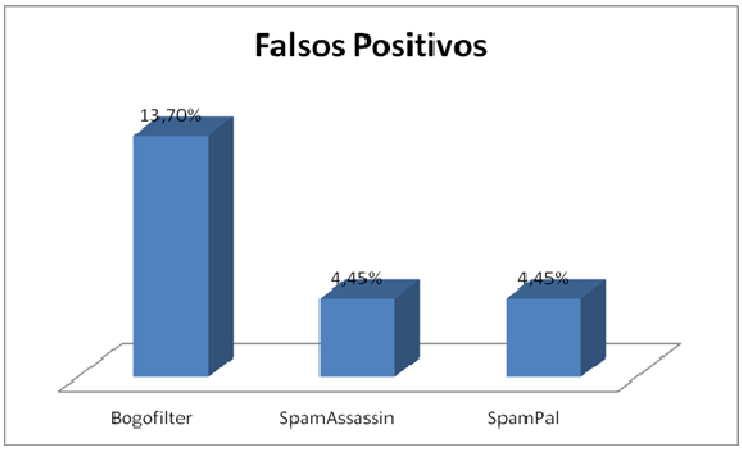

Figura 13 - Gráfico comparativo de incidência de falsos positivos

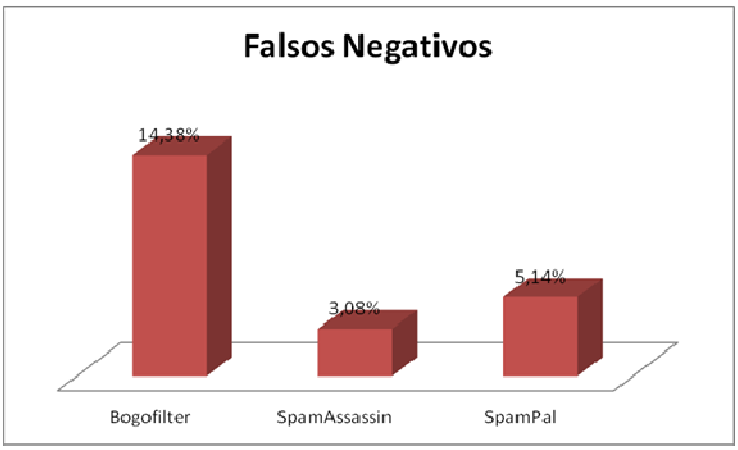

Figura 14 - Gráfico comparativo de incidência de falsos negativos

Com o intuito de melhorar o desempenho do *Bogofilter* na classificação das mensagens os parâmetros do *ham_cutoff* e do *spam_cutoff* foram alterados com base na verificação dos índices de "bogosidade" registrados no resultado do teste de acordo com o que é mostrado a seguir:

```
Antes:
#ham_cutoff = 0.45          # default
#spam_cutoff= 0.99          # default

Depois:
ham_cutoff = 0.50           # default
spam_cutoff= 0.95           # default
```

Figura 15 - Alteração do *ham_cutoff* e *spam_cutoff*

Depois de feitas as alterações na configuração do *Bogofilter* e refeito um novo teste para verificação do resultado obtido, tem-se os dados descritos na tabela que segue.

Tabela 2 - Resultado do teste após reconfiguração do *Bogofilter*

Ferramenta	Acertos Spam	Acertos Ham	Falsos Positivos	Falsos Negativos
Bogofilter	49,48%	73,85%	10,96%	10,27%

No gráfico a seguir é possível comparar cada um dos testes e verificar a melhora no resultado em cada um dos itens analisados.

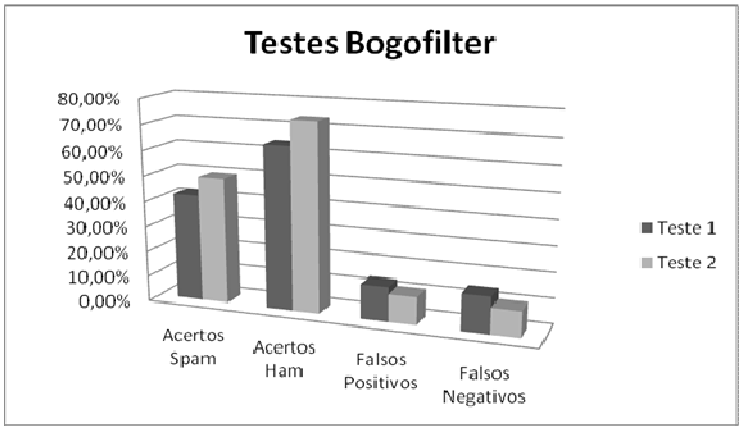

Figura 16 - Gráfico comparativo dos testes com o *Bogofilter*

4. CONCLUSÃO

Diante do exposto no artigo, foi possível constatar a importância de se adotar técnicas e fazer o uso de ferramentas *anti-spam*, visando diminuir os malefícios causados por esse tipo de ataque.

Para se obter um bom resultado na filtragem dos e-mails, a adoção de varias técnicas combinadas e também ferramentas tanto no servidor como no cliente de e-mail, conduz a um resultado mais satisfatório. O uso de *blacklists* e a gerência da porta 25 são exemplos dessas situações.

Com relação aos testes realizados com as três ferramentas de código aberto, *Bogofilter*, *SpamAssassin* e *SpamPal*, constatou-se que em sua configuração *default* o *SpamAssassin* apresentou uma eficácia maior em relação as outras duas em dois dos quatro quesitos analisados, detecção de não *spam* e ocorrência de falsos negativos, e empatando com o *SpamPal* na ocorrência de falsos positivos.

No primeiro teste, utilizando configuração *default*, pode ser constatado que o *Bogofilter* teve o menor desempenho das ferramentas analisadas. Com a alteração dos índices de "bogosidade", foi constatado uma melhora significativa nos quatro quesitos analisados.

É importante que a ferramenta seja treinada com número elevado de mensagens, quanto mais, melhor e mais correta será a filtragem das mensagens, destacando a importância de diminuir ao máximo a ocorrência de falsos positivos, pois deixar de receber um e-mail legítimo é bem pior que receber um *spam*.

Como trabalhos futuros, seria interessante a implementação de um servidor de e-mails em uma pequena ou média empresa, fazendo uso das técnicas estudadas e das ferramentas testadas.

5. REFERÊNCIAS BIBLIOGRAFICAS

BOGOFILTER. Bogofilter FAQ. Disponível em: <http://bogofilter.sourceforge.net/faq.shtml#bogo-what>. Acesso em: 20 outubro 2011.

FABRE, R. C. *Métodos Avançados para Controle de Spam*. Disponível em: <http://www.las.ic.unicamp.br/paulo/ teses/20050215-MP-Recimero.Cesar.Fabre-Metodos.avancados.para.controle.de.Spam.pdf>. Acesso em: 23 novembro 2011.

GRAHAM, P. *A Plan For Spam*. Disponivel em: <http://www.paulgraham.com/spam.html>. Acesso em: 25 outubro 2011.

KUROSE, J. F.; ROSS, K. W. Redes de Computadores e a Internet. São Paulo: Pearson, 2009.

SPAMASSASSIN. *SpamAssassin Configuration Generator*. Disponivel em: <http://www.yrex.com/spam/ spamconfig.php>. Acesso em: 21 novembro 2011.

SPAMPAL. *SpamPal Introdution*. Disponivel em: <http://spampal.sanesecurity.com/manual_eng/intro.htm>. Acesso em: 29 novembro 2011.

ZUCCO, J. C. *Técnicas e Ferramentas de Código Aberto Para Combate ao Spam*. Disponível em: < http://lib.seven. com.br/arquivos/893/JeronimoZucco.pdf>. Acesso em: 15 de outubro de 2011.