

Um modelo para filtragem de mensagens aplicado a uma arquitetura de combate a SPAMs

Isabela L. de Oliveira, Adriano M. Cansian.

UNESP – Universidade Estadual Paulista – Instituto de Biociências, Letras e Ciências Exatas (IBILCE) - Campus de São José do Rio Preto, SP - Brazil

ACME! Computer Security Research

{isabela,adriano}@acmesecurity.org

Abstract. *This paper shows the impact of SPAMs in the daily of the electronic mail users and in the Internet communication infrastructure and it proposes a new model of messages filtering. The more used contention methodologies are analyzed, their characteristics, advantages and disadvantages. A new model of messages filtering based on the frequency distribution of the present characters in its content and in the generation of signatures is described. A combat architecture of Phishing Scam and SPAM is proposed aim to contribute for the contention of the e-mail frauds attempts.*

Resumo. *Este artigo mostra sucintamente o impacto dos SPAMs tanto no cotidiano dos usuários de correio eletrônico quanto na infra-estrutura de comunicação da Internet e propõe um novo modelo de filtragem de mensagens. São abordadas as metodologias de contenção mais utilizadas atualmente, suas características, vantagens e desvantagens. Um novo modelo de filtragem de mensagens, baseado na distribuição de frequência dos caracteres presentes em seu conteúdo e na geração de assinaturas é descrito. Uma arquitetura de combate a Phishing Scam e SPAM é proposta, visando contribuir para a contenção das tentativas de fraudes por e-mail.*

1. Introdução

O envio de mensagens eletrônicas entre pessoas data dos primórdios da comunicação entre computadores. O protocolo responsável pelos *e-mails* trafegados na Internet foi inicialmente proposto em 1980 como MTP [RFC 772]. Desde então tem sido aprimorado e é atualmente representado pelo SMTP [RFC 2821].

Os *e-mails* representam atualmente um valioso mecanismo de comunicação. Nos últimos tempos este mecanismo tem sido utilizado não apenas para facilitação do transporte de mensagens, mas para disseminação de informações não solicitadas e tentativas de fraudes eletrônicas. Este uso indevido tem causado diversos problemas aos usuários da Internet e à sua própria infra-estrutura distribuída de comunicação. Usuários são afetados ao não receber novas mensagens devido à lotação de suas caixas, perda de tempo em sua identificação e eliminação além da sujeição às tentativas de fraudes, como envio de páginas de instituições clonadas e programas que objetivam a obtenção de dados pessoais dos usuários. No âmbito de infra-estrutura, este mau uso pode impactar na banda utilizada, má utilização dos equipamentos dedicados aos serviços de correio além de requerer investimentos para contra medidas.

Os chamados SPAMs e *Phishing Scams* se tornaram mensagens comuns àqueles que se utilizam da comunicação eletrônica. SPAM refere-se às mensagens não solicitadas enviadas em massa para grandes listas de destinatários. São compostas por diversos assuntos, mas predominantemente por conteúdos de caráter comercial. O *Phishing Scam* é uma modalidade de SPAM cujo objetivo é lesar diversos destinatários, enviando falsos anúncios, mensagens de caráter apelativo, levando o usuário a instalar programas maliciosos ou persuadi-lo a visitar páginas maliciosas que assim o façam.

O número deste tipo de mensagem tem se mantido alto nos últimos anos. Diversos laboratórios e grupos monitoram esta atividade mundialmente, como por exemplo, o *SPAMCOP* [Spamcop 2008]. No Brasil, o *Cert.br* tem divulgado estatísticas acerca do abuso de correio eletrônico [Cert.br 2008].

Este artigo descreve uma arquitetura de filtragem de mensagem que possibilita tanto a detecção de mensagens indesejadas (SPAMs) quanto tentativas de fraude (*Phishing Scams*). Primeiro são apresentadas as metodologias atuais para combate ao abuso do serviço de correio eletrônico. Em seguida, o modelo de análise de mensagem é descrito, a arquitetura na qual foi estabelecido, suas características e resultados obtidos durante testes de um protótipo. Por fim, são apresentadas as conclusões.

2. Metodologias atuais de filtragem de mensagens

Em 1975 Jon Postel atentou para o problema de abuso no serviço de correio no documento [RFC 706], referindo-se aos primeiros protocolos de troca de mensagem. Todavia, o problema perdura. Como abordado em [Leavitt 2007], o problema continuará até que o retorno aos *spammers* (financeiro, ou de qualquer espécie) seja muito baixo quando comparado aos riscos de violação de leis ou aos custos para realização da tarefa. Em [Gomes et al., 2004] os autores analisam características do tráfego gerado pelos SPAMs, como a distribuição temporal das mensagens legítimas e ilegítimas, o tamanho das mensagens, o número de destinatários e remetentes, etc. Entretanto, resultados como a distribuição do tamanho das mensagens se modificaram devido à rápida evolução das estratégias dos *spammers*. Em [Andreolini et al., 2005] endereços eletrônicos gerados aleatoriamente são divulgados a fim de contaminar as listas dos *spammers*. Entretanto, não analisam o tempo de permanência dos endereços divulgados, o tempo entre a coleta e o envio das mensagens, e quantas máquinas executam estes processos.

As metodologias de combate às mensagens indesejáveis atuais baseiam-se em protocolos e na filtragem de conteúdo. As baseadas em protocolos verificam se estão de acordo com os padrões de Internet enquanto as filtragens procuram classificar segundo determinadas regras. A seguir, são discutidas as técnicas mais utilizadas na atualidade.

2.1 Greylisting

Este método utiliza a combinação de três informações: o endereço IP do host que está enviando a mensagem, o endereço do remetente e do destinatário no envelope da mensagem. A regra resume-se a “Se a tripla nunca foi vista, então recuse a mensagem e todas as outras que contiverem a mesma tripla, dentro de um determinado período de tempo, com um erro temporário”. Servidores de correio válidos tentarão reenviar as mensagens recusadas, o que não acontece com a maioria dos *spammers*. No entanto, isso implica atraso na recepção de *e-mails* de “novos” remetentes e esse atraso pode ser significativo dependendo das configurações dos MTAs de origem e destino. A

vantagem é não requerer esforços por parte dos usuários além de simplicidade na manutenção, sendo implementado no nível do MTA (*Mail Transfer Agent*). Como desvantagem, spammers que utilizam reenvio no caso de erros podem passar por filtros Greylisting.

2.2 Sender Policy Framework

O SPF (*Sender Policy Framework*) [RFC 4408] é uma política que combate a falsificação do endereço de retorno no envelope de uma mensagem. Para isto, verifica o domínio do remetente de cada e-mail. Sabendo o domínio, através de uma consulta DNS (serviço de resolução de nomes), é possível determinar se o servidor que está enviando a mensagem é realmente autorizado para o domínio em questão. Sendo responsável, pode-se aceitar a mensagem. Caso contrário, o e-mail possivelmente foi forjado. O SPF depende da aceitação e implantação nos MTAs, a fim de tornar a política efetiva. Além disso, é totalmente factível a aquisição de um domínio e conseguinte publicação de SPF e disseminação de SPAMs por parte de um *spammer*, atuando como um servidor válido.

2.3 Domain Keys [Yahoo 2006]

Este mecanismo trabalha com o conceito de chave pública e privada, sendo a privada de posse do MTA e a pública disponível via DNS. Assim o MTA remetente assina suas mensagens possibilitando ao receptor checar, por meio da assinatura, se a origem da mensagem é válida para o domínio em questão. Embora efetiva, esta metodologia possui as mesmas desvantagens do SPF. Ainda, uma grande desvantagem é a carga computacional exigida para a verificação das mensagens, principalmente em servidores carregados, onde o impacto é bastante grande.

2.4 Razor e Distributed Checksum Clearinghouse

A metodologia *Razor* [Razor 1999] muito se assemelha ao modelo descrito neste artigo. A filtragem ocorre baseada em um *hash* gerado para uma mensagem considerada SPAM, sendo que servidores podem trocar informações por meio da distribuição de *hashes*. Sua eficiência está ligada a colaboração de cada cliente para identificação e geração de *hashes*. Ainda, a geração destas assinaturas não considera toda a mensagem.

A metodologia *Distributed Checksum Clearinghouse - DCC* utiliza o mesmo mecanismo da *Razor*, mas independe da colaboração de clientes. Um servidor DCC é responsável por receber a notificação de SPAM (*hashes*) dos clientes colaborativos e calcular quantas delas ocorrem em cada mensagem. Um MTA que use esta política pode consultar um DCC a fim de verificar a quantidade de envios da mensagem recebida, entre os colaboradores. Esta metodologia apresenta problemas com mensagens do tipo *newsletter*, enviadas a diversos endereços.

2.5 Real Time Blacklist

As RBL são listas que identificam em tempo real *hosts* que fazem SPAM ou que fizeram em um período recente. Uma desvantagem é a classificação generalizada de um domínio. Uma vez que um único usuário tenha abusado do serviço de correio incluindo o domínio em uma RBL, todos os outros usuários sofrerão a possibilidade de negação de suas mensagens, caso o MTA destino utilize políticas de checagem de RBLs.

2.6 Rule-Based Filter

Estes filtros realizam classificações baseados em regras de pontuação. Uma vez que a pontuação atinja um limiar pré-definido a mensagem será considerada SPAM. Um dos mais conhecidos filtros *rule-based* é o *Spamassassin* [Spamassassin 2008]. Mecanismos como algoritmos genéticos, redes neurais e classificação segundo probabilidades (*Bayesian Filters*) são metodologias de filtragem. Uma desvantagem é a alta carga de recurso computacional exigida, principalmente em servidores de grande porte.

3. Um novo modelo de filtragem de mensagens

O modelo proposto visa realizar a análise de uma mensagem com base em seu conteúdo. A principal motivação vem da metodologia utilizada por alguns sistemas de detecção de intrusos, denominada *fingerprint*. Nestes sistemas, uma intrusão é codificada na forma de uma assinatura, uma estrutura que representa concisamente um ataque. No modelo proposto cada mensagem possui uma representação própria, de acordo com uma distribuição de frequência dos caracteres presentes em seu conteúdo.

Durante o projeto do modelo, duas características foram tomadas como base para seu desenvolvimento. Uma refere-se ao desempenho computacional exigido durante uma análise, visto que o principal problema de analisadores de conteúdo é a grande exigência de recurso de processamento. A segunda trata de obter uma metodologia de análise de conteúdo sem infringir a privacidade dos usuários. Por fim, detectar SPAM de maneira efetiva e com baixo custo é o propósito do modelo proposto.

Uma mensagem SMTP é composta basicamente pelo cabeçalho e corpo da mensagem. Nesta análise não consideramos o envelope, etapa do diálogo SMTP. No cabeçalho são definidas características como tipo de codificação, remetente e destinatário, etc. No corpo é inserido o conteúdo da mensagem, seja diretamente ou na forma de anexos.

```

From spammer@spam.com Sat Aug 20 00:00:00 2008
Return-Path: <otherspammer@spam.com>
... Outros campos
MIME-Version: 1.0
Content-Type: multipart/related;
  type="multipart/alternative";
  boundary="====_NextPart_000_0000_BDE06FC4.E1911767"
... Outros campos
This is a multi-part message in MIME format.

====_NextPart_000_0000_BDE06FC4.E1911767
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
... Corpo da mensagem

```

Figura 1. Uma mensagem MIME multipart: exemplo do cabeçalho MIME.

O protocolo SMTP possui uma limitação de representação de símbolos, fixada à codificação ASCII [RFC 1345] de 7 bits. Isto é solucionado pelo padrão MIME (*Multipurpose Internet Mail Extensions* - [RFC 2045]). O MIME provê mecanismos para conversão de codificações facilitando o envio de outros tipos de informações, como imagens, caracteres não ASCII, etc. Uma mensagem com este padrão possui um cabeçalho MIME indicando a versão utilizada e o tipo de conteúdo daquela mensagem. Existem diversos tipos MIME. No modelo proposto foram considerados os tipos

multiparted com textos e imagens, os tipos *text* e *image*, e as mensagens em texto plano, que não usam MIME. Estes tipos são declarados no cabeçalho MIME dentro do campo *Content-Type*. A figura 1 mostra um exemplo de mensagem MIME.

3.1 Modelo de assinatura

Quando uma mensagem é analisada, uma assinatura pode ser gerada para representá-la. A assinatura é uma distribuição de frequência dos caracteres que ocorrem em uma mensagem, considerando do ASCII 33 ao 96 (caracteres especiais, algarismos, pontuação e alfabéticos de 'A' a 'Z' maiúsculos) e do ASCII 123 ao 126 (também caracteres especiais). O intervalo de 97 a 128 compreende os caracteres alfabéticos de 'a' a 'z' minúsculos, sumarizados de maneira não sensível à caixa. Estes caracteres não são diferenciados dos maiúsculos, sendo indiferente a ocorrência de 'A' ou 'a', possibilidade essa considerada a mesma ocorrência de caractere na assinatura. Compõem ainda a assinatura, uma soma do total de caracteres e quantos caracteres diferentes ocorrem. A figura 2 é um exemplo de como gerar esta estrutura, utilizando a linguagem C.

```

line: uma linha da mensagem;
text_sig.counter[i]: estrutura que guarda as ocorrências dos caracteres;

for (i=0; i<strlen(line); i++){
    if (line[i] >= 33 && line[i] <=96) text_sig.counter[line[i]-31]++;
    else{
        if (line[i] >= 97 && line[i] <=122) text_sig.counter[line[i]-63]++;
        else{
            if (line[i] >= 123 && line[i] <=126) text_sig.counter[line[i]-31]++;
        }
    }
}

```

Figura 2. Geração de uma assinatura: análise linha a linha.

Na geração de cada assinatura o cabeçalho principal não é considerado. No corpo da mensagem, dois tipos idênticos de assinaturas podem ser gerados: assinaturas que descrevam campos do tipo *text*, e codificação MIME de uma imagem. O modelo proposto gera uma assinatura baseada na codificação MIME da imagem anexa, possibilitando o reconhecimento desta em mensagens posteriores.

Os sistemas de reconhecimento baseados em assinaturas têm problemas quanto a ocorrência de falso-positivos. Neste modelo esta possibilidade é bastante remota. Usando probabilidades podemos obter dados referentes ao poder representativo do modelo. Considerando apenas as possíveis combinações de caracteres diferentes, o modelo é capaz de representar 2^{68} combinações.

Embora haja muitas possibilidades de combinações, a eficiência do modelo está ligada à possibilidade de classificações erradas, que depende do quão diferentes são os caracteres obtidos em uma mensagem. A possibilidade de o sistema cometer um erro está vinculada à possibilidade de duas mensagens possuírem exatamente a mesma combinação de caracteres. Esta probabilidade pode ser calculada segundo

$$P = \frac{1}{C_{68}^x}$$

onde P é probabilidade, C_{68}^x é combinação 68 em x e x é a quantidade de caracteres diferentes obtidos em uma mensagem. Calculando esta probabilidade para $1 \leq x \leq 68$ teremos valores muito baixos no intervalo de 3 a 65 caracteres e alguns casos extremos para a ocorrência de 1, 2, 67 e 68 caracteres diferentes.

A segunda parte do modelo visa amortizar este caso e conferir uma alta capacidade de diferenciação de assinaturas armazenando o número de ocorrências de cada caractere. Assim, para duas assinaturas serem consideradas parecidas, elas devem conter os mesmos caracteres e um número de ocorrências semelhantes. O modelo analisará a ocorrência de cada caractere buscando discrepâncias que as caracterizem como não representantes da mesma mensagem, ou seja, encontre uma determinada distância entre as assinaturas. Na figura 3, duas assinaturas com a mesma quantidade de caracteres diferentes e quantidade total próxima são comparadas. Apesar de haver semelhança quanto aos caracteres, há uma diferença quanto às ocorrências deles.

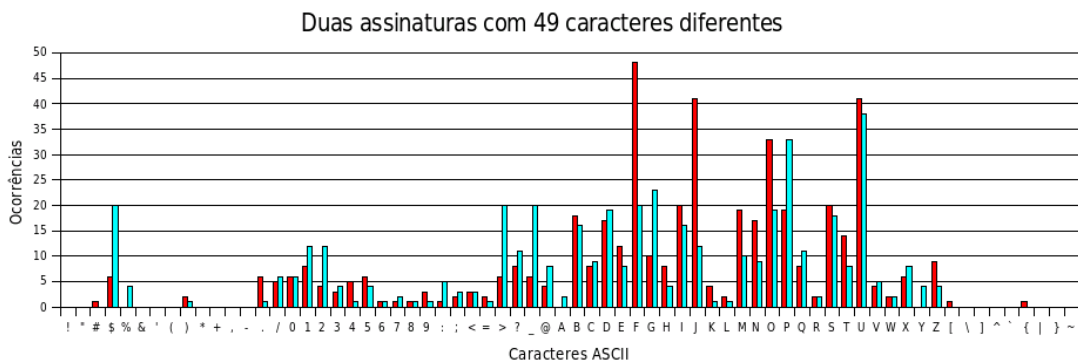


Figura 3. Duas assinaturas de 49 caracteres: diferentes ocorrências para cada um possibilitam uma diferenciação.

Ao comparar as ocorrências dos caracteres, é calculado um fator de representação entre o número de ocorrências na assinatura gerada pela mensagem e o número de ocorrências de uma assinatura previamente conhecida. Assim, para cada caractere é determinada uma distância relativa de ocorrências multiplicada por um fator de $1/68$, resultando no quão esta diferença contribui para a diferença total das assinaturas. Este procedimento pode ser descrito por

$$\sum_{i=1}^{68} \left(1 - \frac{\text{menor}}{\text{maior}} \right) \times \frac{1}{68}$$

onde i representa cada caractere e *menor* e *maior* representam, respectivamente, o menor e o maior valor entre as ocorrências de um mesmo caractere nas assinaturas comparadas. Por exemplo, se em uma assinatura temos 10 ocorrências de um caractere e comparamos com uma assinatura que possua 9 teremos $(1 - 9/10) \times (1/68) = 0.00147058$, uma contribuição baixa visto a semelhança de ocorrências. Ao final da verificação de todos os caracteres, este processo de diferenciação resulta em uma distância de 0.35.

3.2 Arquitetura de implantação do modelo

Nesta seção será abordada a arquitetura de implantação do modelo de filtragem, também utilizada nos testes realizados. Considerando os tipos de filtragem com base nas etapas necessárias para a transferência de uma mensagem, o modelo é classificado como um filtro após enfileiramento (*after queue*) [Postfix 2008]. No modelo descrito, as

mensagens filtradas já se encontram enfileiradas, prontas para ser entregues às caixas dos usuários. A figura 4 mostra a seqüência de passos dentro de um servidor.

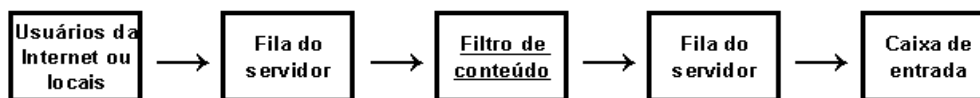


Figura 4. Filtragem após enfileiramento: mensagens são recebidas pelo MTA e posteriormente filtradas.

Esta arquitetura possibilita a utilização do filtro de frequências junto a diversos outros tipos de filtros. Desta forma, filtros podem ser utilizados tanto antes do enfileiramento quanto após a mensagem ser recebida, de acordo com as necessidades e políticas do servidor. O diagrama a seguir demonstra a organização dos componentes desta arquitetura.

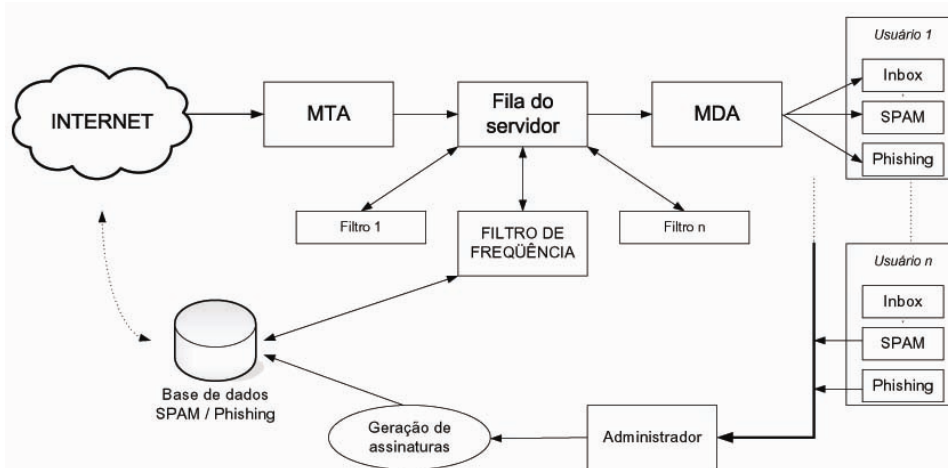


Figura 5. Arquitetura do servidor de correio: filtro após inserção na fila e possibilidade de compartilhamento de base de dados.

Inicialmente uma mensagem é recebida pelo MTA (*Mail Transfer Agent*) e colocada em uma fila para posterior entrega. Uma vez nesta fila, a mensagem deve passar por todos os filtros configurados, sendo colocada novamente na fila após todas as verificações. Neste momento, o MDA (*Mail Delivery Agent*) é responsável por retirá-la da fila e entregá-la a caixa de mensagens do usuário destino.

Nesta arquitetura o MDA tem papel fundamental. É a partir dele que as mensagens são separadas em válidas e inválidas (SPAMs ou *Phishings*) e entregues aos usuários. Durante o processo de filtragem, se uma mensagem é classificada como inválida, uma linha é inserida no seu cabeçalho a fim de que o MDA possa diferenciá-las. O protótipo utilizado insere uma linha semelhante à *X-MAILFILTERSTATUS: Yes - Differ 0.225488*, indicando que a mensagem foi identificada com alguma assinatura do banco, com uma distância de 0.225488. Assim como mencionado, foi utilizado um limiar de 0.15 para que esta classificação fosse possível. Esse valor foi obtido pela análise das assinaturas da caixa SPAM, assim este limiar é suficiente para que estas mensagens sejam classificadas como SPAM. Esta marcação permite que o MDA faça a separação das mensagens e entregue ao usuário separadamente das mensagens válidas.

Em relação à manutenção do sistema, esta pode ser efetivada de maneira colaborativa. Uma vez que o usuário tenha concordado com as classificações ou mesmo

inserido novas mensagens que classifiquem como SPAMs ou *Phishings*, estas podem passar por uma seleção, realizada por um administrador, sendo utilizadas na geração de assinaturas a popularem a base de dados.

Por fim, estas informações da base de assinaturas podem ser compartilhadas entre diversos servidores, de acordo com políticas de atualizações e operação. Ainda, diversos órgãos gestores na Internet, como o CAIS (Centro de Atendimento a Incidentes de Segurança) [Cais 2008] e os Certs (*Computer Emergency Response Team*) respondem rapidamente a eventos de fraudes disponibilizando informações que podem ser inseridas nas bases e propagadas de forma bastante rápida.

4. Resultados: testes realizados com um protótipo

A fim de testar as características do modelo proposto foi desenvolvido um protótipo capaz de integrar-se ao MTA Postfix [Postfix 2008] e a um banco de dados, responsável pelo armazenamento das assinaturas. Este protótipo foi desenvolvido na linguagem C, devido à geração de binários extremamente rápidos, contribuindo assim para o desempenho da filtragem.

Os testes foram realizados com base em mensagens acumuladas, separadas em duas caixas denominadas SPAMs e INBOX. A caixa SPAM é composta totalmente por mensagens inválidas, enquanto a INBOX possui em sua maioria, mensagens normais. A tabela 1 mostra as características de cada um desses conjuntos.

Tabela 1. Conjuntos de mensagens utilizados para testes.

Conjuntos					
	Total de mensagens	Inválidas	Válidas	Com imagens	Tamanho
SPAM	4545	4545	0	1175	41 MB
INBOX	1376	7	1369	25	102 MB

A primeira etapa de testes teve como objetivo verificar o comportamento do sistema segundo o desempenho computacional. Para tanto, foram utilizadas duas plataformas diferentes de hardware. Primeiramente foram testados os tempos necessários para a leitura de um conjunto de mensagens, geração das assinaturas e inserção em uma base de dados. O segundo teste consistiu em utilizar o mesmo conjunto (SPAM) apenas para a classificação das mensagens, sendo necessária, para cada mensagem do conjunto, sua leitura, geração de assinatura, busca das assinaturas parecidas na base de dados e comparação para determinação da distância entre elas. Neste teste também podemos verificar a quantidade de erros do tipo falso-negativo (classificar uma mensagem inválida como válida). O terceiro teste foi realizado usando um conjunto de mensagens majoritariamente válidas (INBOX). Neste caso, obtemos a quantidade de erros do tipo falso-positivo (mensagens válidas que são classificadas como inválidas). A tabela 2 mostra os tempos médios obtidos para cada teste.

Tabela 2. Teste de desempenho envolvendo duas plataformas de hardware: tempo para inserção e verificação de conjuntos de mensagens.

	Velocidade do processador	Memória dos sistemas	Tipo de disco (E/S)	Tempo médio de inserção da caixa SPAM	Tempo médio de verificação da caixa SPAM	Tempo médio de verificação da caixa INBOX
Hardware 1	800 Mhz	758 MB	ATA 100	23s	3min48s	1min15s
Hardware 2	3 Ghz	1 GB	SATA	13s	1min24s	30s

Podemos verificar que o sistema leva pouco tempo para inserir as assinaturas no banco de dados e para classificar as mensagens como SPAM ou não. Esses resultados mostram o alto desempenho computacional que o sistema possui.

Por fim, foram analisadas as taxas de erros cometidas pelo sistema. De acordo com os conjuntos utilizados, duas medidas foram possíveis: taxa de falso-negativo no conjunto SPAM e taxa de falso-positivo no conjunto INBOX. Das 4545 mensagens avaliadas no conjunto SPAM, obtivemos 6 falso-negativos, o que representa uma taxa de erro de 0.13%. Ainda, estas mensagens foram verificadas como casos específicos, nas quais o corpo da mensagem é vazio. Neste caso, o sistema não aceita a representação por assinaturas, o que levaria a uma estrutura nula que pode ocorrer com frequência em mensagens válidas. Todas as mensagens que possuíam imagens foram corretamente detectadas. Isto mostra tanto a efetividade de sistemas de reconhecimento baseados em assinaturas quanto a do modelo de distribuição de frequência.

No caso do conjunto INBOX, das 1376 mensagens analisadas obtivemos 9 falso-positivos, representando uma taxa de 0.65%. Destes erros, 7 foram devido a classificação errada de imagens e 2 devido a classificação errada da parte textual da mensagem. Este conjunto é composto por mensagens diversas. Possui ainda, 7 mensagens inválidas que foram detectadas corretamente pelo sistema. Esses resultados mostram a eficiência do sistema para classificar as mensagens. Ainda, o sistema teve um alto desempenho ao ter uma baixa taxa de falso-positivo e falso-negativo.

A integração com o MTA Postfix mostrou-se funcionalmente correta. As mensagens recebidas pelo MTA são passadas a um controlador (via UNIX Pipe) e filtradas pelo sistema, sendo inserida uma linha em seu cabeçalho indicando o *status* da mensagem. A mensagem é então devolvida à fila do MTA e entregue ao usuário, de acordo com seu *status*. Na arquitetura proposta anteriormente, foi utilizado o MDA Procmail [Procmail 2008], responsável pela entrega diferenciada das mensagens (separação nas caixas de SPAM e Inbox).

5. Conclusões

A aceitabilidade do termo SPAM entre os usuários da Internet demonstra o quão voraz é a disseminação destas mensagens. No entanto, a complacência com tal prática dada pela ausência de mecanismos proibitivos efetivos, como leis punitivas ou controladoras, têm impactado fortemente no atual mundo digital. O número de fraudes e desfalques financeiros tem aumentado consideravelmente. Neste contexto, as medidas de contenção e minimização deste problema têm se tornado cada vez mais importante.

Este trabalho demonstra a importância das mensagens não solicitadas no cotidiano dos usuários da Internet e descreve a proposta de um modelo de filtragem aplicado a uma arquitetura de combate ao SPAM. Diversas metodologias são atualmente utilizadas para a mitigação dos efeitos destas mensagens. O modelo proposto apresenta características de várias metodologias, podendo ser comparado a filtros *Razor* devido ao caráter colaborativo e baseado em assinaturas, aos analisadores probabilísticos e aos baseados em regras de conteúdo. Ele traz importantes características, como bom desempenho, não possibilidade de quebra de privacidade e detecção de *Phishings* cadastrados sem possibilidade de falso-negativo. Sua arquitetura visa possibilitar sua implantação em servidores de grande escala, reduzindo o grau de abrangência de uma fraude.

Finalmente, o sistema de classificação é baseado em limiares definidos pelo administrador. Nos testes realizados foram utilizados valores definidos empiricamente, mesmo assim culminando em resultados que demonstram efetividade. Trabalhos futuros presumem metodologias mais apuradas para ajuste destes valores, bem como a utilização de algoritmos diferentes de distanciamento e sistemas *Fuzzy* de classificação, diferenciando a importância de cada parte componente de uma mensagem.

Referências

- Andreolini, M., Bulgarelli, A., Colajanni, M. e Mazzoni, F. (2005) “Honeyspam: Honeypots fighting spam at the source”. In: SRUTI05: Steps to Reducing Unwanted Traffic on the Internet Workshop, p. 77 – 83.
- Cais. (2008) “Cais: Centro de Atendimento a Incidentes de Segurança”, <http://www.rnp.br/cais>, Maio.
- Cert.br. (2008) “CERT.br. Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil”, <http://www.cert.br/stats/spam>, Maio.
- Gomes, L. H., Cazita, C., Almeida, J. M., Almeida, V., Wagner Meira, J. (2004) “Characterizing a spam traffic”. Em ACM SIGCOMM conference on Internet measurement (IMC’04). Pages: 356 – 369. ACM Press.
- Harris, E. (2008) “Greylisting: the next step in the spam control”, <http://projects.puremagic.com/greylisting/>, Maio.
- Leavitt, N. (2007) “Vendors Fight Spam’s Sudden Rise”. Computer, Vol.40, Iss.3, March 2007, Pages:16-19.
- Postfix. (2008) “Postfix: Postfix After-Queue Content Filter”, http://www.postfix.org/FILTER_README.html, Maio.
- Procmal. (2008) “Procmal mail processing”, <http://www.procmal.org/>, Maio.
- Postel, J. (1975) “RFC 706: On the junk mail problem”, <http://www.faqs.org/rfcs/rfc706.html>, Novembro.
- Sluizer, S. e Postel, J. (1980) “RFC 772: Mail Transfer Protocol”, <http://www.faqs.org/rfcs/rfc772.html>, Setembro.
- Simonsen, K. (1992) “RFC 1345: character Mnemonics and Character Sets”, <http://www.faqs.org/rfcs/rfc1345.html>, Junho.
- Freed, N. e Borenstein, N. (1996) “RFC 2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, <http://www.faqs.org/rfcs/rfc2045.html>, Maio.
- Klensin, J. (2001) “RFC 2821: Simple Mail Transfer Protocol”, <http://www.faqs.org/rfcs/rfc2821.html>, Abril.
- Wong, M. e Schlitt, W. (2006) “RFC 4408: Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail”, <http://www.faqs.org/rfcs/rfc4408.html>, Maio.
- Spamassassin. (2008) “Spamassassin: The Apache SpamAssassin Project”, <http://spamassassin.apache.org/>, Maio.
- Spamcop. (2008) “Spamcop.net”, <http://www.spamcop.net/spamstats.shtml>. Julho.
- Razor. (1999) “Razor: spam should not be propagated beyond necessity”, <http://razor.sourceforge.net>. Abril.
- Yahoo! Inc.(2006) “Domainkeys: Proving and protecting e-mail sender identity”, <http://antispam.yahoo.com/domainkeys>. Maio.