Técnicas de detecção e classificação de *spam*

Franklin Eufrásio da Silva, Wagner Toscano
Uninove. São Paulo – SP [Brasil]
franklin.silva@gmail.com

Neste artigo, serão abordadas as técnicas de detecção, avaliação e classificação da mensagem eletrônica não-solicitada enviada em massa (*spam*), com ênfase nas técnicas de análise que empregam inteligência artificial (IA) e redes, que interagem compartilhando informações sobre a origem desses *e-mails* pela internet. Três cenários serão utilizados, com o intuito de apresentar uma comparação entre as técnicas bayesiana, filtro com base em assinaturas, *greylist* e DNSBL(*domain name system black list*).

Palavras-chave: *Anti-spam*. *E-mail*. Internet.

1 Introdução

Spam, mensagem eletrônica não-solicitada enviada em massa, é hoje uma realidade que representa quase 50% do total de *e-mails* enviados. O conteúdo varia desde a venda de produtos, boatos, propagandas, golpes financeiros, até a disseminação de *softwares* maliciosos, como os vírus, *worms* e *trojans*.

O *spam* é economicamente viável, pois o remetente não tem custos operacionais com o gerenciamento de *e-mails*. Além disso, os controles de segurança adotados para o bloqueio desse tipo de mensagem existentes no mundo são pouco implementados.

Em alguns países, existem leis específicas que classificam o *spam* como campanhas não solicitadas, puníveis com multas ou rescisões financeiras. Para mitigar os prejuízos causados pelo *spam*, empresas e especialistas desenvolvem técnicas tanto para detecção quanto para o tratamento de mensagens não-solicitadas, possibilitando que inúmeras empresas e usuários tenham o controle sobre os *e-mails* realmente produtivos.

Com base nas pesquisas já realizadas e diante das novas frentes de desenvolvimento sobre inteligência artificial (IA) aplicada ao reconhecimento de *spams*, neste artigo intenta-se aprofundar e analisar as técnicas de detecção mais utilizadas tanto no meio acadêmico quanto no privado. Para isso, comparam-se, a partir de dados experimentais, a eficiência e a relação custo-benefício de cada técnica, permitindo definir as melhores alternativas para a detecção e a classificação de *spams*.

1.1 Técnicas de detecção de spam

Há diferentes técnicas para detectar e classificar *spam*. Os mais simples são detectados e bloqueados por meio do remetente; para outros, utilizam-se técnicas de análise de conteúdo que possibilitam averiguar padrões para identificação e caracterização. Essas técnicas, entretanto, tendem a ter um custo de processamento maior, pois examinam todo o conteúdo de uma mensagem.

Muitos e modernos sistemas *anti-spam* utilizam técnicas de aprendizado por meio de IA, o que aumenta consideravelmente sua eficiência. Porém, alguns usuários podem achar intrusivo saber que determinadas ferramentas estão sendo adotadas para ler e/ou analisar o conteúdo de seu *e-mail*.

Como exemplo de implementação das técnicas de detecção de *spam*, há *softwares* do tipo "cliente de *e-mail*" que interagem diretamente com o usuário, classificando as mensagens conforme suas preferências pessoais e, por exemplo, de acordo com assuntos que ele pré-determinou (DIAS, 2005).

Neste artigo, aborda-se a eficiência das técnicas de tratamento de mensagens legítimas que foram classificadas como *spam* (falsos-positivo) e do tipo *spam* que não foram detectadas devido a um erro de análise da mensagem (falsos-negativo).

Outras implementações atuam diretamente em servidores de *e-mail*, que, de forma similar a um cliente de *e-mail* intrusivo, não só classifica como também detecta se uma mensagem pode ser considerada *spam*. No entanto, uma implementação em servidor precisa ser eficiente para processar inúmeras mensagens simultaneamente.

Serão comentadas as implementações de técnicas com base em filtros e análise heurística, como a do serviço global de busca por listas de *spams* (ou *domain name system black list*, também chamada DNSBL), filtro com base em conteúdo, *greylist*, sistemas de pergunta e resposta e filtro bayesiano.

1.2 DNSBL

O DNSBL é utilizado como filtro heurístico e bloqueio.

O serviço DNSBL faz a publicação de listas (geralmente endereços de *internet protocol* [IP]),

de tal forma que, se um *spam* estiver cadastrado em um DNSBL, qualquer servidor de *e-mail* poderá encontrá-lo por meio de consultas a um DNSBL, possibilitando, então, classificar uma mensagem como um *spam* (TIPTON; KRAUSE, 2004).

Existem inúmeros servidores DNSBL disponíveis na internet. Cada um deles trata de categorias específicas de *spams*, ou seja, o usuário de um serviço de DNSBL deve escolher qual servidor se enquadra na categoria de *spam* que deseja pesquisar. Como exemplo, é possível consultar um que possui somente o cadastro de *spammers* (remetente de um *e-mail* não-solicitado ou malicioso) de redes de banda larga (redes que comumente são utilizadas para envio de *spams*) ou um de *hosts* da internet que, em algum momento, "abusaram" da quantidade de *e-mails* enviados em um curto período de tempo (prática também comum de um *spammer*).

A técnica DNSBL requer três itens para classificar um *spam*: domínio, servidor de domínio e lista de endereços para publicar os *spammers*.

É possível criar um serviço DNSBL utilizando o *software* livre *Bind*, que é um servidor DNS popular.

Para fazer uma consulta a um servidor DNSBL, é necessário configurar o servidor de *e-mail* ou o *software anti-spam* que vem nele instalado, adicionando quais servidores DNSBL serão consultados. Dessa forma, no momento da chegada de cada *e-mail*, esse *software* fará uma pergunta a todos os servidores DNSBL listados a respeito de algumas características desse *e-mail*, como "você, conhece esse remetente?"; "esse remetente é válido?"; "a origem desse *e-mail* é legítima?".

Eis o funcionamento: quando o *software anti-spam* recebe uma resposta positiva, ou seja, a indicação de que o *e-mail* em questão pertence a uma lista de *spammers*, a pesquisa pode retornar um registro do tipo TXT, contendo o motivo de o endereço IP (Internet Protocol) pesquisado estar nessa lista. Feito isso, o *software* decide se o *e-mail* deve ser classificado ou não como *spam*.

1.3 Filtro de conteúdo

Até recentemente, técnicas de filtro com base em conteúdo utilizavam palavras cadastradas pelos operadores ou administradores das ferramentas *anti-spam*. Dessa forma, se um servidor de *e-mail* recebesse um *spam* contendo a palavra "Viagra", o administrador da ferramenta deveria adicioná-la à configuração, para que o servidor rejeitasse qualquer mensagem que a contivesse. Porém, modernos filtros de conteúdo podem também executar testes adicionais, como analisar o cabeçalho de um *e-mail*.

Segundo Fabre (2005), *spammers* podem ser inseridos em endereços falsos no cabeçalho de uma mensagem com a intenção de esconder as identidades. Além disso, há muitas formas de manipulação de um cabeçalho, que servem de base para a realização da análise.

São grandes as desvantagens de um filtro de conteúdo: o consumo elevado de recurso de processamento e alto número de falsos-positivo. Um administrador de sistemas que rejeite um *spam* por meio desse filtro pode acabar recusando algum outro *e-mail* válido.

Finalmente, os *spammers* podem modificar as frases ou a forma de escrever uma determinada palavra, utilizando técnicas de inserção de hifens ou espaços entre uma sílaba e outra. Como exemplo, se o *spammer* emprega a palavra "Viagra", como "V1agra" ou "Via_gra", torna-se difícil para o analisador de conteúdo identificar essa mensagem como um *spam*.

1.4 Filtro bayesiano

O filtro bayesiano foi proposto, inicialmente, por Mehran Sahami . Ele analisou um documento

por um sistema de classificação, popularizado e proposto por Paulo Graham, que criou a chamada "classificação bayesiana" para detectar e julgar se uma mensagem é *spam* ou não.

Uma vez configurados, os filtros bayesianos não requerem manutenção. Por outro lado, usuários devem marcar mensagens como *spam* ou não-*spam*, e o *software* de filtro aprenderá de acordo com essas marcações, criando uma base de dados com conhecimento de todos os *e-mails* processados até aquele momento. Dessa forma, o filtro bayesiano não reflete as técnicas de programação utilizadas pelo programador ou administrador do *software*.

A análise heurística é utilizada em muitos *softwares anti-spam*, como o SpamAssassin, que adota todas as técnicas de análise mencionadas neste artigo e, adicionalmente, a análise heurística, responsável por sondar cada mensagem com a finalidade de detectar anomalias que fujam de padrões da base de conhecimento. Grande parte dos sistemas desse tipo de análise emprega DNS (Domain Name System) como base para suas pesquisas na internet.

Fabre (2005) afirma que a técnica de análise heurística permite grande flexibilidade quanto à classificação de um *e-mail*, por meio de origem, apesar de essa técnica tender a utilizar pouco recurso de processamento.

A vantagem de um analisador bayesiano é a rápida aprendizagem quanto à constante mudança de palavras, de forma automática, sem intervenções administrativas.

Spammers tentam, constantemente, enganar filtros bayesianos inserindo caracteres estranhos junto a palavras, fazendo com que um analisador bayesiano classifique o *spam* como uma mensagem autêntica.

Alguns *softwares* livres, que também implementam filtros bayesianos, são: Bogofilter, SpamAssassin e o programa de *e-mail* Thunderbird, da Mozilla. Um outro projeto interessante é o CRM114, que lança mão de *hashes*[1] de frases e classificação bayesiana de mensagens.

Utilizando algoritmos de filtro bayesiano, alguns *softwares* como o POPFile fazem categorização automática de *e-mails*, que lhes permite reconhecer uma mensagem como "familiar" ou "de trabalho".

A probabilidade direta de uma hipótese H condicionada a um corpo de dados E, P (H|E) está relacionada ao inverso da probabilidade dos mesmos dados e sujeita à hipótese H, P (E|H).

Matematicamente: $P(H|E) = P(E|H) \cdot P(H)$

O algoritmo utilizado neste artigo teve como base a implementação do filtro bayesiano para a área da saúde (FABRE, 2005), aplicando-se o mesmo conceito, mas adaptado para o ambiente de detecção de *spam*.

Definem-se as variáveis: probabilidade (P); hipótese (H); corpo de dados (E).

Exemplo: Classificação de *spam*

H = *spam* contendo vírus.

E = mensagem com tamanho conhecido (indica a presença de um vírus).

Os dados utilizados pelo *software anti-spam* foram:

$P(H|E) = 0,5$

$P(H) = 1/1000$

$P(E) = 1/50$

$P(H|E) = P(E|H) \cdot P(H)$

$P(E) = 0,5 \cdot (1/1000) = 0,0005 = 1/50$

A probabilidade de um *spam* conter vírus, levando-se em consideração uma mensagem de tamanho conhecido, é de 0,05.

1.5 Filtro com base em assinaturas

Um filtro que se baseie na análise de assinaturas (leia-se também conteúdo das mensagens) possui a vantagem de classificar uma mensagem enviada, analisando se ela é idêntica à que foi recebida. Além disso, quando o texto de uma mensagem contiver o nome ou o *e-mail* com assinaturas, esse filtro poderá identificar facilmente se a mensagem é um *spam* ou não, pois não só um administrador de sistemas poderá fazer esse reconhecimento, como também o próprio usuário, por meio da simples comparação entre as assinaturas.

A técnica com base em assinaturas constantemente atualizadas emprega uma rede distribuída que contém um catálogo de cada *spam* detectado por essa rede, sendo possível realizar consultas simples que poderão retornar um *spam* conhecido seguido de uma assinatura.

Uma mensagem é validada por meio de sua reputação (classificação de uma mensagem como legítima, possivelmente um *spam*, ou algo parecido) e, dessa forma, ela é classificada na rede distribuída.

A implementação da técnica de assinatura é feita por meio de múltiplos filtros que encapsulam partes de um *e-mail*. Para tanto, pode-se utilizar os algoritmos de geração de *hashes* SHA1 de 256 *bits* ou do tipo Ephemeral.

O *software* livre de detecção de *spam* conhecido como *Razor* implementa a técnica de reconhecimento com base em assinaturas.

A implementação é simples: o *Razor* faz uma solicitação a uma rede distribuída na internet a respeito da mensagem em análise. Caso a rede informe que se trata de um *spam*, esse *software* adiciona o cabeçalho *X-Razor-Spam* à mensagem, dotando-a dessa classificação. A seguir, pode ser visto o cabeçalho completo de uma mensagem que foi classificada como *spam* pelo *software Razor*.

Received: from unknown (HELO mail01.zeronet.com.br) ([200.174.111.253]) (envelope-sender <espaco@espacoaberto.com>) by 0 (qmail-ldap-1.03) with SMTP for <nazareth@nazareth.com.br>; 15 May 2006 19:16:43 -0000

Received: from ESPAÇO ABERTO <espaco@espacoaberto.com> by mail01.zeronet.com.br (Zeronet Mail Server) with ASMTP id WYG23348 for <nazareth@nazareth.com.br>; Thu, 15 May 2006 16:14:48 -0300

From: ESPAÇO ABERTO <espaco@espacoaberto.com> To: nazareth@nazareth.com.br Subject: TREINAMENTO “MULHER: PEGUE AS RÉDEAS DE SUA CARREIRA!”

Sender: ESPAÇO ABERTO <espaco@espacoaberto.com>

Mime-Version: 1.0

Content-Type: text/html; charset=”iso-8859-1”

Date: Thu, 15 May 2006 19:14:57 GMT

X-Mailer: ICS SMTP Component V2.32

X-Razor-Spam: SPAM

X-Qmail-Scanner-Message-ID: <11479798039247393@server01>

*** Qmail-Scanner Quarantine Envelope Details Begin *** X-Qmail-Scanner-Mail-

From: “espaco@espacoaberto.com” via server01 X-Qmail-Scanner-Rcpt-To: nazareth@nazareth.com.br

X-Qmail-Scanner: 1.25st (clamdscan: 0.88/1247. spamassassin: 3.1.0. perlscan: 1. 25st. problem Found. Processed in 1.002064 secs) process 7393

Quarantine-Description: SPAM exceeds „quarantine“ threshold - hits=7.0/5.0

SA_REPORT hits = 7.0/5.0 1.0 SUBJ_ALL_CAPS Subject is all capitals

0.0 UNPARSEABLE_RELAY Informational: message has unparseable relay lines

0.1 BR_CURSO_BODY BODY: Curso no body

2.0 BR_SPAMMER_URI URI: Texto suspeito

0.4 MAILTO_TO_REMOVE URI: Includes a ‚remove‘ email address

0.0 HTML_MESSAGE BODY: HTML included in message

3.5 BAYES_99 BODY: Bayesian spam probability is 99 to 100% [score: 1.0000]

0.0 MIME_HTML_ONLY BODY: Message only has text/html MIME parts

1.6 Sistemas de pergunta e resposta

Outra técnica também utilizada por alguns provedores de internet tem objetivo de detectar e classificar *spams*. Para isso, submete o remetente do *e-mail* ao teste conhecido como sistemas de pergunta/resposta.

Um exemplo é o sistema empregado pelo provedor de internet Universo *On-line* (UOL), por meio de sua técnica *anti-spam*. Nele, o usuário recebe uma pergunta na forma de uma imagem contendo uma palavra ou frase, para que sejam reproduzidos por ele no campo específico.

A proposta dessa solução é assegurar que sistemas automáticos (como robôs virtuais) sejam incapazes de ler ou processar imagem.

Em sua dissertação de mestrado, Fabre (2005) define:

> Críticos dessa técnica fazem diversas questões quanto a eficácia, como o exemplo de as pessoas com deficiência visual não poderem ver a imagem e, conseqüentemente, não conseguir interagir com o sistema *anti-spam*, ou então problemas causados por sistemas de pergunta/resposta em grupos de discussões *on-line*, fazendo com que uma mesma pergunta seja enviada a todos os membros de um grupo podendo trazer graves conseqüências a um servidor de *e-mail*.

Alguns usuários que utilizam o sistema de pergunta/resposta reportam sua extrema eficiência quanto ao bloqueio de *spams*, pois esse sistema obriga que cada remetente confirme o envio do *e-mail*.

A implementação da técnica pergunta/resposta é simples e eficiente, pois para cada *e-mail* recebido pelo usuário, é enviado um *e-mail* de pergunta ao remetente para confirmar o envio, fazendo com que *spammers* rejeitem essa pergunta. Somente um remetente legítimo responderá à mensagem para o usuário final.

O sistema, a partir da resposta, enviará automaticamente a mensagem original e com autenticidade para a caixa postal do usuário receptor.

1.7 *Greylist*

Greylist é uma técnica utilizada para bloquear *spams* de forma transparente ao usuário. Seu funcionamento baseia-se na postergação da entrega da mensagem, empregando, para tanto, o *simple mail transport protocol* (SMTP) – sistema sincronizado de comunicação para trafegar mensagens em meio eletrônico.

Quando um *e-mail* é recebido em um servidor, a mensagem é temporariamente rejeitada, retornando como "tente novamente mais tarde" – o que ocorre no protocolo de *e-mail* STMP (TIPTON; KRAUSE, 2004). Ele é armazenado em um banco de dados que contém um conjunto de informações suficientes para identificar, unicamente, cada mensagem. Em um curto intervalo de tempo, servidores de STMP, que aderem corretamente aos padrões do protocolo, farão uma nova tentativa de envio. Ao receber novamente a mensagem para tentar mais tarde, o servidor remetente irá reenviá-la.

Ao receber a mensagem reenviada, o servidor que utiliza a técnica *greylist* pesquisará na base de dados o histórico da mensagem rejeitada. Obtendo resposta positiva, ela será liberada e chegará ao destinatário.

O protocolo de *e-mail* SMTP é considerado de transporte não-confiável pelo fato de não checar a identificação e a autenticidade do usuário que o está utilizando; portanto, a possibilidade de falhas temporárias está embutida em seu núcleo.

Segundo Souto (2004), todo servidor de *e-mail* bem implementado promove tentativas de en-

trega de uma mensagem que tenha obtido um código de falha temporária. Isso geralmente ocorre, por exemplo, quando a fila de um servidor-destino está muito longa para ser processada, ou o servidor tem uma carga muito alta (de operações de entrada/saída). É nesse aspecto que o *greylisting* é bem-sucedido. Na Figura 1, pode ser visualizado o funcionamento dessa técnica:

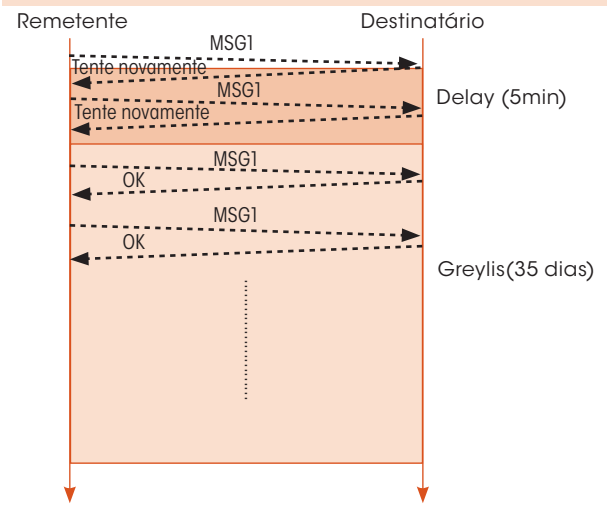

Figura 1: Esquema da técnica *greylist*.
Fonte: O autor.

Para ser viável, um *spammer* utiliza um servidor de *e-mail* desenvolvido para desconsiderar mensagens de erro no destino. Por exemplo, servidores de *e-mail* recebem uma carga de mensagens que devem ser entregues para usuários gerados a partir de uma lista de nomes. Para o *spammer*, processar cada mensagem com falha geraria um custo alto, pois consumiria largura de banda e recursos computacionais.

Também deve ser considerada a eficácia de novos *softwares* utilizados por *spammers* que, muitas vezes, podem detectar a técnica *greylisting* por meio da análise de tempo de resposta, ou até mesmo possuindo filas de *e-mail* que simulam um servidor de *e-mail* padrão, que pode gerenciar mensagens não entregues e reenviar o *spam* ao usuário final.

2 Metodologia

A implementação de todas as técnicas de detecção e o controle de *spams* utiliza *software* de código aberto e de conhecimento público.

Para realizar a análise, foi necessária uma grande quantidade de *e-mails* pré-classificados em *spam* e não-*spam*.

Os *e-mails* considerados *spams* foram coletados de diversas fontes públicas na internet e de uma conta de correio criada pelo autor dessa pesquisa. Ao todo, foram adquiridas 1.015 mensagens.

Para a base de *e-mails* considerados não-*spams*, foram utilizadas mensagens legítimas enviadas por colaboradores de algumas listas de discussão, além da base de mensagens do autor desse documento.

O principal objetivo da apresentação dos resultados foi analisar a eficiência e eficácia dos algoritmos para determinar a aplicabilidade de cada um, agindo isoladamente ou em conjunto.

Faz-se importante lembrar que, na realização dos testes, todas as técnicas apresentaram taxas de falsos-negativo e falsos-positivo, comum nos sistemas descritos no artigo.

3 Resultados

Foi analisada a performance de algoritmos de aprendizado no contexto de filtros *anti-spam* e estáticos, além de técnicas de DNS e redes de compartilhamento de listas de *spams*. O aumento no volume de *spams* tem gerado a necessidade de utilização desses algoritmos e técnicas.

Foram criados três cenários específicos para permitir resultados aleatórios, variando, em cada cenário, o número de testes e atributos, tais como quantidade de *e-mails* e de *spams* autênticos, tempo de processamento de cada *e-mail* e infor-

mações estatísticas relacionadas a falsos-positivo e falsos-negativo.

A grande vantagem obtida quanto a esses fatores evidenciou-se durante a análise do filtro bayesiano, pois por ser treinado, possui a característica de categorizar os *spams* recebidos. O ponto negativo desse tipo de filtro está relacionado a um mau treinamento, ocasionando falsos-positivo, ou seja, mensagens legítimas sendo classificadas como *spams*, trazendo sérios problemas ao remetente e ao destinatário. Porém, o filtro pode ser corrigido, resultando em melhorias no aprendizado.

Falsos-positivo têm maior impacto negativo em comparação a falsos-negativo. Devido à classificação de um *e-mail* legítimo como *spam* falso-positivo o tempo de detecção desse tipo de erro e a recuperação desse *e-mail* levam um certo tempo.

3.1 Cenários utilizados nos testes

São três os cenários envolvidos para a execução dos testes e obtenção dos resultados com base na utilização isolada do filtro bayesiano, filtro bayesiano + *greylist* e, por fim, nas combinações das técnicas filtro bayesiano e DNSBL; pergunta/resposta e filtro com base em assinaturas.

3.1.1 Resultados do primeiro cenário de testes

No mês de março de 2006, foi focada a aplicação isolada do filtro bayesiano que possibilitou analisar seus pontos positivos e negativos quanto à classificação de todos os 1.015 *e-mails*.

Na Tabela 1, no total de *e-mails* considerados como não-*spam*, somente dois foram classificados como *spams*, totalizando 0,3% de erro. Porém, percebe-se uma dificuldade do filtro bayesiano em classificar mensagens que são *spams*, totalizando 5,7% de erro.

	Falsos-positivo	Falsos-negativo	*E-mails*	%
Spam	0	18	315	5,7
Não-spam	2	0	700	0,3

Tabela 1: Classificação de *spams* (filtro bayesiano)
Fonte: O autor.

A assimilação do filtro bayesiano foi eficiente somente após um longo período de aprendizado, seguindo um padrão de acertos com nível razoável.

3.1.2 Resultados do segundo cenário de testes

Além da facilidade e portabilidade dessas soluções, o conjunto utilizado pelo filtro bayesiano e DNSBL mantém a popularidade em constante crescimento, graças à existência de muitas implementações de *software* livre (já citado neste artigo). Em razão disso, prova-se ser útil e acessível a muitos usuários de *e-mail*.

Na Tabela 2, a técnica DNSBL resulta em uma média de acerto maior, destacando-se pelo alto grau de atualização dos servidores DNSBL.

Técnica	Acertos (%)
DNSBL + filtro de conteúdo	97,3

Tabela 2: Acertos de *spam* com utilização isolada
Obs.: Estatística gerada no mês de março de 2006
Fonte: O autor.

Nos testes desse cenário, isoladamente, essa técnica apresenta significativa eficiência por ter como característica a consulta a listas de *spammers*, compartilhadas via internet, o que possibilita a obtenção de informações sobre *spams*, quase em tempo real, ao contrário da técnica de filtro bayesiano (ou "filtro de conteúdo"), que exige intervenção manual na ferramenta, estando, portanto, suscetível a erros.

3.1.3 Resultados do terceiro cenário de testes

Combinando as técnicas de análise bayesiana com a de estática ou consultando listas públicas, é possível determinar o comportamento e a relação de custo/benefício.

Seguem, na Tabela 3, os resultados dos testes utilizando a combinação das técnicas de análise bayesiana com as de assinaturas, DNSBL e pergunta/resposta.

Técnica	Acertos (%)	Erros (%)
Assinatura + filtro bayesiano	93,5	6,5
Filtro bayesiano + pergunta/resposta	99,7	0,3
Assinatura + pergunta/resposta	98,5	1,5
Pergunta/resposta + DNSBL	97,3	2,7
Filtro por conteúdo + pergunta/resposta	98,1	1,9
Assinatura + DNSBL	91,8	8,2
Assinatura + pergunta/resposta + DNSBL	92,8	7,2

Tabela 3: Acertos de *spam* com utilização combinada
Obs.: Estatística gerada no mês de março de 2006.
Fonte: O autor.

Na Tabela 3, o combinado de técnicas da linha três obteve bons resultados quanto à detecção de *spams*, com 98,5% de acertos, em média. Quanto à utilização conjunta das técnicas Assinatura e DNSBL, a primeira evita que a segunda classifique erroneamente algumas mensagens legítimas como *spam*, mesmo sem contribuir para o grau de acerto. Além disso, essa combinação impede que a análise seja feita pelo DNSBL, que consome mais recursos de *hardware* que a técnica de pergunta/resposta, e proporciona, inclusive, uma economia de *links* de dados, não sendo necessário que toda a mensagem seja entregue ao servidor destino para ser analisada.

Nas demais combinações, a análise apresentou resultados muito próximos quanto ao acerto de classificação das mensagens como *spam*, apesar da ocorrência de um percentual maior de falsos-positivo.

Já em relação à combinação filtro bayesiano e pergunta/resposta, os resultados chegaram a 99,7% de acertos, o que indica baixa margem de erros.

4 Considerações finais

O *spam* estimula o desenvolvimento de técnicas cada vez mais avançadas para detecção e classificação, o que contribui para o uso racional dos recursos de *hardware* e internet e, conseqüentemente, para o aumento de produtividade.

Algumas técnicas necessitam de uma base de treinamento, outras apenas são aplicadas no cabeçalho do *e-mail* ou na transação SMTP, quando o servidor remetente envia uma mensagem ao servidor destinatário.

Atualmente, instituições acadêmicas ou empresas, ao ter um servidor de *e-mail* conectado à internet, obrigatoriamente precisam utilizar técnicas e ferramentas instaladas e configuradas a fim de evitar *spam*, com o mínimo de falsos-positivo. Isso demanda pesquisa e conhecimento técnico avançado.

Com base nos resultados, conclui-se que a combinação da técnica de análise bayesiana com a de pergunta/resposta foi a mais eficiente na detecção de *spams* na configuração padrão, porém com um número muito alto de falsos-positivo. Assim, conclui-se que a necessidade de treinamento constante dessa ferramenta é imprescindível, e o uso inicial deve ser acompanhado de outras técnicas como a análise de assinatura, *greylist* e DNSBL.

A técnica de análise de assinaturas, que inclui análise de conteúdo, foi a que apresentou a alternativa mais interessante para o treinamento, pois

os usuários finais dos servidores de *e-mail* participam da aprendizagem da ferramenta por meio da troca de *hashes* de cada mensagem.

A combinação da técnica de assinatura com a de DNSBL obteve o pior resultado. Um dos fatores que contribuíram para isso foi o número de regras utilizadas (algumas para detecção também na língua portuguesa). O incremento de números de regras, como no *software* SpamAssassin, resultou em grande custo computacional.

Com base nos resultados obtidos, é perceptível que as técnicas analisadas neste artigo possuem um bom nível de eficiência quando utilizadas em conjunto, mas que não há um algoritmo ou técnica com 100% de eficiência, pois, constantemente, *spammers* criam novas formas de burlar os sistemas *anti-spam*.

É curioso o fato de, para cada nova técnica *anti-spam*, criar-se também uma maneira de burlá-la. Esse é um problema histórico do protocolo de *e-mail* SMTP. Ao ser elaborado em agosto de 1982, não havia a necessidade de desenvolver mecanismos de segurança, pois não era previsto que a ferramenta de *e-mail* fosse utilizada para envio de *spams*, vírus e *worms*.

Com base nesses problemas, a pesquisa deve seguir novo horizonte, orientada na criação de outro protocolo de *e-mail* capaz de garantir a autenticidade do usuário, de forma mandatória, aliada às campanhas de educação e de conscientização quanto à utilização das ferramentas de *e-mail*, pois somente a tecnologia não será suficiente para conter os *spams*.

Detection and classification techniques of SPAM

In this paper, we discussed techniques of detection, evaluation and classification of non-requested eletronic messages massively sended (*spam*), with emphasis on the analysis technique that applied Artificial Inteligence (AI) and networks, which interacts, sharing informations about the origin of these emails through the internet. Three sceneries were used, in order to show a comparison among Baysean, Filter based on signatures, greylist and DNSBL techniques.

Keyword: Anti-spam. E-mail. Internet.

Notas

1 N. Ed.: O termo inglês *hash*, em segurança de computadores, é um número gerado a partir de uma seqüência de caracteres, usado para garantir que a mensagem seja transmitida mantendo a sua integridade intacta.

Referências

DIAS, S. P. *Sistema anti-spam baseado em greylists*. Universidade Federal de Lavras. Departamento em Ciência da Computação. Lavras: UFLA, jul 2005.

FABRE, R. C. *Métodos avançados para controle de spam*. 2005. Dissertação (Mestrado em Computação [Rede de Computadores])-Laboratório de Administração e Segurança de Sistemas, Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2005.

SOUTO, M. C. P. de. *Naïve bayesian learning*. Universidade Federal do Rio Grande do Norte. Departamento de Informática e Matemática Aplicada. Natal: Dimap-UFRN, 2004. Disponível em: <http://www.dimap.ufrn.br/~marcilio/IA/IA2004.1/naive-bayes.ppt>. Acesso em: 4 abr. 2006.

TIPTON, H. F.; KRAUSE, M. *Information security management handbook*. 5. ed. Danvers: Auerbach, 2004

Recebido em 23 jun. 2006 / aprovado em 2 set. 2006

Para referenciar este texto

SILVA, F. E. da; TOSCANO, W. Técnicas de detecção e classificação de *spam*. *Exacta*, São Paulo, v. 4, n. 2, p. 333-342, jul./dez. 2006.