UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ELETRÔNICA E DE COMPUTAÇÃO

Análise de Técnicas de Defesa contra *Spam*

Autor:

Danilo Michalczuk Taveira

Orientador:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.Ing.

Examinadores:

Prof. Luís Henrique Maciel Kosmalski Costa, Dr.

Prof. Marcelo Gonçalves Rubinstein, D.Sc.

Prof. Marcelo Luiz Drumond Lanza, M.Sc.

DEL

Dezembro de 2006

À minha família.

Agradecimentos

Aos meus pais e demais familiares pelo carinho, compreensão e incentivo ao longo de toda a graduação.

Aos amigos da graduação, André, Almeida, Bruno, Felipe, Kaio, Mariangela, Marília, Natalia, Neimar, Pantoja e Pietro por todos os anos que convivemos.

Aos amigos do GTA/COPPE, Carina, Diana, Gustavo, Igor, Marcelo, Miguel, Pedro, Reinaldo e Zuquim pela contribuição e incentivo para a realização deste projeto.

Ao meu orientador, professor Otto Duarte e aos professores Luís Henrique e Marcelo Rubinstein pelos conselhos durante a execução deste trabalho.

Aos professores do Departamento de Eletrônica e de Computação.

A todos que me incentivaram, contribuindo de forma direta ou indireta, para a minha formação profissional.

Resumo do Projeto Final apresentado ao DEL/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletrônico e de Computação.

Análise de Técnicas de Defesa contra *Spam*

Danilo Michalczuk Taveira

Dezembro de 2006

Orientador: Otto Carlos Muniz Bandeira Duarte
Departamento: Engenharia Eletrônica e de Computação

O envio de mensagens eletrônicas não solicitadas (*spams*) é uma atividade muito lucrativa para os *spammers* e cresce a cada dia. O grande número de *spams* gera um enorme prejuízo tanto para os usuários quanto para provedores de serviço e operadoras de redes. Uma forma de combater os *spams* são os mecanismos anti-*spam* que implementam um conjunto de regras e procedimentos com o objetivo de identificar e filtrar os *spams*. No entanto, identificar os *spams* e criar técnicas para combatê-los não é uma tarefa trivial, uma vez que as técnicas utilizadas pelos *spammers* para enviarem as mensagens evoluem de forma a se adaptar e contornar os mecanismos anti-*spam*.

Nesse trabalho, as técnicas utilizadas para criar e enviar *spams*, assim como a evolução dos procedimentos usados pelos *spammers* para contornar os mecanismos anti-*spam* são descritos. Os principais mecanismos anti-*spam* existentes e propostos na literatura são apresentados e avaliados. A avaliação do desempenho dos mecanismos é feita medindo-se os falsos positivos e os falsos negativos.

Para se efetuar uma análise das principais características dos mecanismos anti-*spam* mais utilizados foi desenvolvido um sistema denominado ADES (**A**nálise **DE** **S**pam). A arquitetura do sistema ADES é modular e flexível, permitindo a análise de diferentes características das mensagens e do processo de envio de *spams*. Os resultados obtidos da análise mostram a baixa eficiência dos mecanismos atuais de combate aos *spams* e a necessidade de novos mecanismos, que não sejam baseados em análise por conteúdo, para combater as novas formas de *spams*.

Palavras-Chave

Mensagens Não Solicitadas

Spam

Redes de Computadores

Mecanismos anti-*spam*

Lista de Acrônimos

BGP :	*Border Gateway Protocol*;
CAPTCHA :	*Completely Automated Public Turing test to tell Computers and Humans Apart*;
CDF :	*Função de Distribuição Cumulativa*;
DMZ :	*Demilitarized Zone*;
DNS :	*Domain Name System*;
DNSBL :	*Domain Name System Blacklist*;
GTA :	*Grupo de Teleinformática e Automação*;
HTML :	*HyperText Markup Language*;
HTTP :	*HyperText Transfer Protocol* ;
IMAP4 :	*Internet Message Access Protocol version 4*;
IP :	*Internet Protocol*;
MX :	*Mail Exchange*;
ORBS :	*Open Relay Behavior-modification System* ;
POP3 :	*Post Office Protocol version 3*;
PTR :	*Pointer*;
RBL :	*Real-time Blackhole List*;
RFC :	*Request For Comments*;
SMTP :	*Simple Mail Transfer Protocol* ;
SPF :	*Sender Policy Framework*;
TCP :	*Transmission Control Protocol.*

Sumário

Resumo **iv**

Lista de Acrônimos **vi**

Lista de Figuras **x**

Lista de Tabelas **xii**

I Introdução **1**

I.1 Objetivos 6

I.2 Organização do Texto 6

II Técnicas de envio de *spams* **7**

II.1 Coleta de dados 7

II.2 Formato das mensagens 10

II.3 Envio das mensagens 11

III Técnicas de combate aos *spams* **14**

III.1 Sistemas baseados em filtragem simples 15

III.1.1 Listas brancas e negras 15

III.1.2 Uso de pesos e regras 17

III.1.3 Filtros bayesianos 19

III.2 Sistemas com auto-aprendizado 21

III.2.1 Caracterização de tráfego de *spams* 22

III.2.2 Listas cinzas 24

III.2.3 Potes de mel 26

III.2.4 Padrões sociais 28

III.3 Sistemas baseados na verificação da origem 31

III.3.1 Verificação do endereço DNS reverso 32

III.3.2 *Sender Policy Framework* (SPF) 34

III.3.3 Desafio e resposta 35

IV Sistema de Análise de *Spam* - ADES **42**

IV.1 Módulo Pote de Mel 43

IV.2 Módulo de verificação dos mecanismos 44

IV.2.1 Listas Negras 47

IV.2.2 *Sender Policy Framework* (SPF) 48

IV.2.3 DNS reverso 48

IV.2.4 Pesos e Regras 50

IV.3 Módulo de análise dos resultados 51

V Testes e análise dos resultados **53**

V.1 Metodologia 53

V.2 Resultados 54

V.2.1 Resultados dos mecanismos 54

V.2.2 Características dos *spams* 57

V.2.3 Análise do pote de mel 61

VI Conclusão **65**

Referências Bibliográficas **70**

Lista de Figuras

I.1 Foto de várias latas de *SPAM*. 2

I.2 Classificação dos *spams* 2

I.3 Sistema de correio eletrônico. 4

II.1 Exemplos de formatação de *spams*. 12

III.1 Verificação de listas negras. 17

III.2 Grafo de um classificador bayesiano. 20

III.3 A estrutura de um pote de mel. 28

III.4 Um exemplo de rede social. 29

III.5 Um exemplo de uma rede social e de uma rede anti-social. 32

III.6 Um exemplo de registro SPF. 35

III.7 Um exemplo de CAPTCHA. 38

IV.1 Servidor de políticas. 45

V.1 Percentual de falsos positivos e falsos negativos. 56

V.2 Distribuição do tempo de verificação de cada mecanismo. 57

V.3 Distribuição temporal das mensagens. 58

V.4 Distribuição por hora das mensagens. 59

V.5 Distribuição por tamanho das mensagens. 60

V.6 Distribuição por faixas de endereços IP das mensagens. 60

V.7 Distribuição geográfica dos endereços IP que enviam *spam*. 62

V.8 Distribuição do tempo entre coleta e envio. 63

V.9 Distribuição dos endereços em função do número de mensagens recebidas. . 64

V.10 Distribuição da distância entre os endereços IP. 64

Lista de Tabelas

III.1 Mecanismos do SPF. 36

III.2 Modificadores do SPF. 37

IV.1 Parâmetros do servidor de políticas do *Postfix*. 45

IV.2 Nome-código dos mecanismos. 47

IV.3 Resultados da biblioteca *Mail::SPF*. 49

V.1 Resultado da análise dos mecanismos. 55

V.2 Países responsáveis pelo envio de *spams*. 61

Capítulo I

Introdução

Com a popularização da Internet, um número cada vez maior de usuários tem acesso aos serviços de correio eletrônico, mensagens instantâneas e voz sobre IP. Nesse contexto, o envio de mensagens eletrônicas não solicitadas (*spams*) é visto como uma atividade lucrativa [1]. Como os *spams* em sua maioria contêm propagandas, grande parte dos custos de divulgação do anunciante são transferidos para os provedores de serviço, responsáveis pelo encaminhamento das mensagens até os destinatários.

A origem do nome *spam* é duvidosa, mas especula-se que o nome escolhido seja o de um produto da empresa Hormel Foods [2] que é um presunto enlatado chamado *SPAM*. A Figura I.1 apresenta uma foto de várias latas de *SPAM*. Esse produto era servido na segunda guerra mundial como principal refeição para os soldados do exército. Depois do final da guerra muitos soldados receberam uma medalha e a medalha ficou popularmente conhecida como medalha *spam*. Além disso, em um esquete do grupo Monty Python existe uma personagem que lê os pratos do cardápio do bar e em todos eles aparece o *SPAM*. Dessa maneira, a palavra *spam*, que era apenas o nome do presunto enlatado, passou a ser sinônimo de algo repetitivo.

A maioria dos *spams* é de conteúdo comercial, anunciando um determinado produto ou serviço, mas também existem outros tipos de *spams* [3]. A Figura I.2 mostra os principais tipos de *spams* e o percentual numérico de cada tipo.

Figura I.1: Foto de várias latas de *SPAM*.

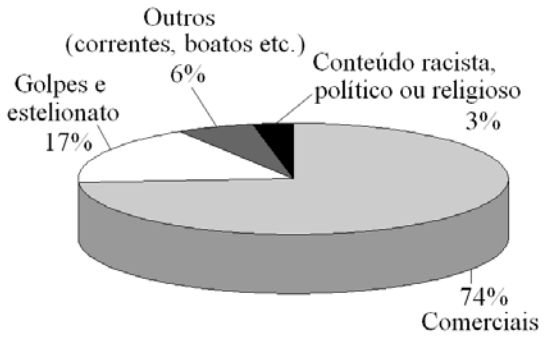


Figura I.2: Classificação dos *spams* [4].

As correntes e boatos têm como objetivo divulgar histórias falsas. Como as pessoas acabam acreditando nas histórias, repassam as mensagens, gerando uma quantidade de mensagens muito grande. As mensagens que tem como objetivo aplicar golpes e estelionato, também chamadas de *scams*, procuram fazer com que os usuários acreditem em uma determinada história e acabem caindo num golpe ou fraude. Os tipos de golpe mais comuns são os que anunciam herança em um país distante e que para receber a quantia da herança de alguns milhões o destinatário da mensagem deve depositar uma determinada quantia em uma conta para pagar as despesas. Muitos usuários acabam caindo nesses golpes e depositam o dinheiro. Existem também muitas tentativas de fraude, onde o *spammer* envia uma mensagem como sendo de um banco ou outra entidade conhecida e pede para que o usuário confirme alguns dados ou passe o número da conta e senha para um suposto recadastramento.

A maior motivação para o envio dos *spams* é o retorno financeiro que é obtido pelos

spammer. Em primeiro lugar, o custo de divulgação é muito menor do que uma divulgação feita através de meios de comunicação convencionais como rádio, TV e meios impressos como jornais ou revistas. Outra vantagem dos *spams* é que podem atingir um número muito grande de usuários e, diferentemente dos outros meios de comunicação convencionais, de todas as partes do mundo. Nesse cenário, os *spammers* usam os *spams* como uma forma altamente lucrativa de anunciar produtos ou serviços. Estima-se que um *spammer* gaste cerca de US$250 para enviar um milhão de *spams* [5]. Em uma pesquisa com usuários a respeito das suas atitudes em relação aos *spams*, 39% dos usuários domésticos e 13% dos usuários corporativos disseram que já compraram produtos ou serviços oferecidos através de *spams* [3]. Outra motivação para o envio de *spams* é o enriquecimento ilícito, através de golpes ou tentativas de estelionato.

A vantagem para os *spammers* da divulgação com um custo reduzido causa uma série de prejuízos, tanto para usuários como para os provedores de serviço e operadoras de telecomunicações. Para os usuários o prejuízo maior está na perda de tempo com a leitura dos *spams*. No meio corporativo, estima-se que cada empresa tenha um prejuízo anual da ordem de US$1300 por cada empregado, devido a perda de produtividade, gastos com tecnologia de combate aos *spams* e gastos com infra-estrutura. Para os provedores de serviço e operadoras de telecomunicações os gastos são principalmente devido ao desperdício de recursos da rede com a transmissão dos *spams*, além dos gastos com o processamento e armazenamento das mensagens, soluções anti-*spam*, treinamento de pessoal, chamadas de suporte e outros gastos indiretos. Estima-se que o prejuízo para os provedores seja da ordem de US$2800 para cada milhão de *spams* encaminhados [6].

Atualmente, o combate aos *spams* é um grande desafio na Internet. Estatísticas apontam que os e-mails não solicitados já representam 68,7% de todas as mensagens que circulam na Internet. Para tentar combater o *spam* existem os mecanismos anti-*spam*, que são compostos de procedimentos e regras que tem como objetivo identificar e filtrar os *spams*, reduzindo a perda de tempo e os gastos com infra-estrutura para armazenamento e transmissão das mensagens. Apesar de todos os esforços para bloqueá-los, muitos ainda não são filtrados devido ao avanço das técnicas para burlar os programas anti-*spam*.

A grande facilidade de se enviar *spams* e a grande dificuldade de combatê-los se deve à simplicidade do protocolo usado no sistema de correio eletrônico da Internet. Este sistema é composto de agentes de usuário, servidores de correio e vários protocolos para a comunicação entre as entidades do sistema. A Figura I.3 mostra um exemplo de um sistema de correio eletrônico entre dois usuários.

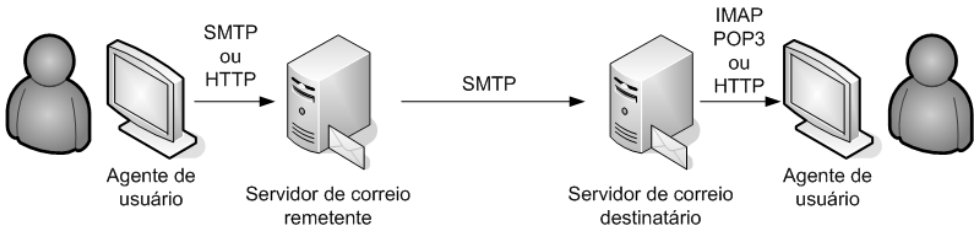


Figura I.3: Sistema de correio eletrônico.

Os agentes de usuário são programas utilizados pelo usuário para enviar e receber mensagens. Os agentes de usuário mais conhecidos são o Microsoft Outlook, o Mozilla Thunderbird e o Eudora. Os servidores de correio têm como objetivo enviar e receber mensagens assim como armazenar as mensagens dos seus usuários. A comunicação entre os agentes de usuário e os servidores de correio eletrônico para receber as mensagens é geralmente feita através dos protocolos POP3, IMAP4 ou HTTP. O protocolo POP3 é um protocolo simples que permite ao cliente ler e apagar as mensagens do servidor. Já o protocolo IMAP4 é mais completo que o protocolo POP3, oferecendo opções para busca de mensagens, gerenciamento de pastas de mensagens no servidor e armazenamento de estado das mensagens no servidor. O protocolo HTTP também é usado quando são utilizados os agentes de usuário pela web, também conhecidos como *webmails*.

Para a comunicação entre os servidores e o envio da mensagem do agente de usuário para o servidor, o protocolo geralmente utilizado é o SMTP[1]. Esse protocolo é definido na RFC 2821 [7] e sua principal característica é ser um protocolo muito simples e com poucas mensagens. O protocolo usa conexões TCP e utiliza como porta padrão a porta 25. O protocolo SMTP original não define nenhum mecanismo de autenticação e verificação da

[1]No caso do usuário utilizar um agente de usuário pela web, será utilizado o protocolo HTTP para mandar a mensagem para o servidor.

validade das informações.

Os principais comandos do protocolo SMTP são: `HELO`, `MAIL FROM`, `RCPT TO`, `DATA`, `QUIT` e `VRFY`. O comando `HELO` é utilizado para identificar a máquina que está enviando a mensagem. Os comandos `MAIL FROM` e `RCPT TO` indicam, respectivamente, o remetente e destinatário das mensagens. Para iniciar o envio do cabeçalho e o corpo da mensagem é utilizado o comando `DATA` e, por fim, utiliza-se o comando `QUIT` para terminar a conexão. O comando `VRFY` pode ser utilizado para verificar se um usuário existe. O exemplo a seguir mostra o envio de uma mensagem, onde S representa o servidor e C o cliente.

```
S: 220 servidor.br
C: HELO cliente.br
S: 250 Hello cliente.br, pleased to meet you
C: MAIL FROM: <usuario@cliente.br>
S: 250 usuario@cliente.br... Sender ok
C: RCPT TO: <usuario@servidor.br>
S: 250 usuario@servidor.br ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: From: usuario@cliente.br
C: To: usuario@servidor.br
C: Subject: Teste
C:
C: Teste de envio de correio.
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 servidor.br closing connection
```

O cabeçalho da mensagem é composto pelas linhas iniciais antes de uma linha em branco e contém informações como o remetente, o destinatário e o assunto da mensagem.

O corpo da mensagem corresponde à parte da mensagem que é mostrada ao destinatário e contém o texto da mensagem.

Já que o protocolo SMTP não garante a validade das informações, no comando `MAIL FROM` um usuário pode colocar qualquer endereço eletrônico, permitindo a falsificação do remetente de forma extremamente fácil. Em tentativas de golpes e estelionato, o *spammer* forja o endereço do remetente de forma a esconder a verdadeira origem e, conseqüentemente, diminuir a desconfiança do usuário sobre a mensagem.Os campos do cabeçalho da mensagem também podem ser alterados.

Para solucionar o problema da autenticação foi criado o mecanismo SMTP-AUTH definido na RFC 2554 [8]. Esse mecanismo, embora possibilite que um provedor realize a autenticação dos seus usuários, não permite que um provedor ao receber uma mensagem tenha certeza de que a mensagem foi originada do endereço de origem que ela possui no seu cabeçalho, já que ele pode ser facilmente falsificado.

I.1 Objetivos

O objetivo deste trabalho é descrever os mecanismos anti-*spam* atuais e propostos na literatura, além de analisar a eficiência dos mecanismos anti-*spams* mais utilizados. Além disso, serão analisadas as características das mensagens *spams* e do processo de envio de *spams*. Para isso foi implementado um sistema chamado ADES (**A**nálise **DE** **S**pam) que possibilita a realização de todos os testes realizados.

I.2 Organização do Texto

Este trabalho está organizado da seguinte forma. No Capítulo II são apresentadas as principais técnicas utilizadas por *spammers* para enviar mensagens. No Capítulo III são analisados os principais mecanismos anti-*spam*. No Capítulo IV é apresentado o protótipo de sistema para avaliação dos mecanismos anti-*spam*. Detalhes referentes aos testes e à análise dos resultados são abordados no Capítulo V. Por fim, no Capítulo VI serão apresentadas as conclusões e as considerações sobre perspectivas futuras.

Capítulo II

Técnicas de envio de *spams*

Nesta seção são discutidas as técnicas utilizadas pelos *spammers* para enviarem os *spams*. Essas técnicas têm como principais objetivos enganar os mecanismos anti-*spam* e atingir o maior número de destinatários possível. O processo para enviar os *spams* é composto de três etapas: coleta de dados, formatação das mensagens e envio. A etapa de coleta de dados tem como objetivo criar uma lista com os destinatários dos *spams*. A etapa de formatação da mensagem se refere à criação do conteúdo das mensagens. Por fim, tem-se o envio das mensagens.

II.1 Coleta de dados

Nessa fase os *spammers* procuram obter o maior número possível de endereços eletrônicos possíveis, para posteriormente servirem de destinatários dos *spams* enviados. O procedimento mais utilizado para a coleta de dados é a varredura de páginas da Internet, de grupos de discussão, de arquivos de listas e de outros meios que possuam diversos endereços eletrônicos. Para que a coleta seja eficiente e de custo muito baixo, o procedimento é totalmente automatizado através de um software chamado robô. O robô acessa uma determinada página, busca em seu conteúdo endereços eletrônicos e segue para uma nova página repetindo o procedimento. A busca é extremamente simples, pois todo endereço

eletrônico deve possuir o caractere @ seguido do domínio. Para tentar dificultar a tarefa dos robôs de busca de endereços eletrônicos, está se tornando comum a substituição do símbolo @ do endereço eletrônico por palavras como `at`, `[at]` e outras variações. A idéia básica é dificultar a tarefa de busca do robô mas, ao mesmo tempo, manter simples a identificação do endereço eletrônico por um ser humano. Para uma pessoa fica razoavelmente claro que se trata de um endereço eletrônico e que deve ser adicionado o símbolo @. Já para os mecanismos robôs essa alteração torna o processo mais complicado, já que agora ao invés de buscar apenas o símbolo @ nas páginas, existem várias combinações diferentes que podem ser usadas. Um mecanismo ainda mais eficaz para enganar os robôs é criar uma figura contendo o endereço eletrônico, o que exige do robô a execução de um processo de reconhecimento de caracteres em imagem para descobrir o endereço eletrônico. Um aspecto importante a ser observado é que mesmo se o usuário não divulgar seu endereço eletrônico em sua página ou utilizar técnicas como as descritas anteriormente, ainda existem sítios que divulgam os endereços eletrônicos sem nenhuma preocupação com os robôs de coleta de endereços eletrônicos.

Outro meio de coleta de dados é através da invasão de sítios ou computadores através de vírus[1], cavalos de tróia[2] ou outras pragas digitais [9]. A invasão de sítios tem a vantagem de obter informações sobre o perfil dos usuários, além do endereço eletrônico. Assim, quanto mais informações do usuário o *spammer* possuir, a propaganda poderá ser mais seletiva e direcionada. As listas construídas também podem ser vendidas para outros *spammers* e quanto mais informações estiverem disponíveis maior o valor comercial dessas listas. Novos vírus também são desenvolvidos com o objetivo de capturar toda a lista de contatos dos usuários e enviar para os *spammers*.

Existem também ataques de dicionário onde as listas são construídas utilizando nomes próprios e palavras comuns [10]. Essa técnica tem uma grande efetividade em domínios com muitos usuários, já que existe uma grande quantidade de usuários que possuem endereços com apenas o seu primeiro nome. Outra técnica similar é a de ataque de força

[1] Trecho de código com o objetivo de se anexar a outros programas e propagar-se.

[2] Programas que permitem acessos não autorizados aos recursos do computador.

bruta, em que são enviadas mensagens para todos os endereços eletrônicos dentro de uma determinada faixa de endereços, como por exemplo, entre aaaaa e zzzzz. Nesse exemplo todas as pessoas que possuem um endereço eletrônico de cinco caracteres irão receber os *spams*. Apesar dessa técnica possuir uma baixa eficiência para domínios com poucos usuários, para domínios grandes muitos usuários irão receber os *spams*.

Uma etapa importante do processo de coleta também é a verificação dos endereços eletrônicos, já que uma lista com um número muito grande de endereços inexistentes vai causar uma perda de tempo para os *spammers*. Um dos meios que eram utilizados pelos *spammers* era o uso do comando `VRFY` do protocolo SMTP. Entretanto, esse comando é desabilitado pela maioria dos administradores, com o objetivo de dificultar a verificação dos endereços. Atualmente a verificação dos endereços é feita incluindo iscas nas mensagens, como um link para a remoção do usuário na lista. O usuário, achando que não vai mais receber os *spams*, ao clicar nesse link acaba apenas confirmando para o *spammer* que aquele endereço existe e, além disso, o usuário lê as mensagens de *spams*. Dessa forma o *spammer* passa a enviar ainda mais *spams* para esse usuário. Outra técnica utilizada para realizar a verificação é a inclusão de imagens na mensagem que contêm um determinado número ou texto de referência relacionado com o endereço eletrônico. Assim, quando o usuário visualizar a mensagem, o seu programa de correio eletrônico irá acessar o servidor onde a imagem está hospedada para buscá-la e enviar esse texto ou número de referência juntamente com a requisição da imagem. Verificando os números ou textos de referência, o *spammer* pode então saber quais são os endereços válidos e ativos. Para combater esse processo os programas de correio eletrônico mais novos não abrem imagens externas por padrão, sendo necessário que o usuário confirme a abertura de imagens externas caso deseje. Entretanto, a falta de informação dos usuários pode fazer com que os usuários sempre confirmem a abertura das imagens.

II.2 Formato das mensagens

No início, a construção das mensagens de *spam* era feita sem nenhuma preocupação com o seu conteúdo. *Spams* eram facilmente identificados pela análise de conteúdo ao se verificar determinadas palavras, como por exemplo: vigra, cialis etc. Assim, para contornar os mecanismos anti-*spam* baseados em análise de conteúdo, os *spammers* passaram a se preocupar com o formato das mensagens, de forma a evitar ou camuflar determinadas palavras.

Uma decisão importante dessa fase é a opção por criar mensagens no formato texto ou mensagens no formato HTML. As mensagens de texto geralmente são mais difíceis de serem classificadas como *spam* pelos mecanismos de análise de conteúdo, já que são geralmente bem curtas. Isso dificulta a identificação de características de que a mensagem se trata de um *spam*. Esse formato, no entanto, tem o inconveniente de ser menos atrativo para o usuário, já que não permite o uso de cores, imagens, animações e outros efeitos que chamam a atenção. O formato HTML fornece uma grande variedade de opções para formatação do conteúdo e permite o uso de figuras. Portanto, o formato HTML permite a criação de uma mensagem muito mais atrativa para o usuário que uma mensagem no formato texto. Os *spammers*, para atrair a atenção do usuário, se servem do formato HTML empregando textos com caracteres coloridos e de diferentes tamanhos além de usarem imagens. A desvantagem para os *spammers*, nesse caso, é que estas mensagens HTML geralmente possuem um tamanho maior e características adicionais que podem ser usadas para facilitar a classificação destas mensagens como *spam* [4].

Por outro lado, a versatilidade do formato HTML também é usada pelos *spammers* para tornar mais difícil a análise do conteúdo do texto por parte dos mecanismos anti-*spam*. Uma técnica muito utilizada é a disposição de palavras em células de uma tabela, onde cada uma das letras da palavra é colocada em uma célula de uma linha ou coluna da tabela. O mecanismo anti-*spam* para poder analisar o conteúdo da mensagem, tem então que interpretá-la e descobrir que as letras que estavam separadas na verdade formam apenas uma palavra. Esse processo, no entanto, aumenta a complexidade da análise de

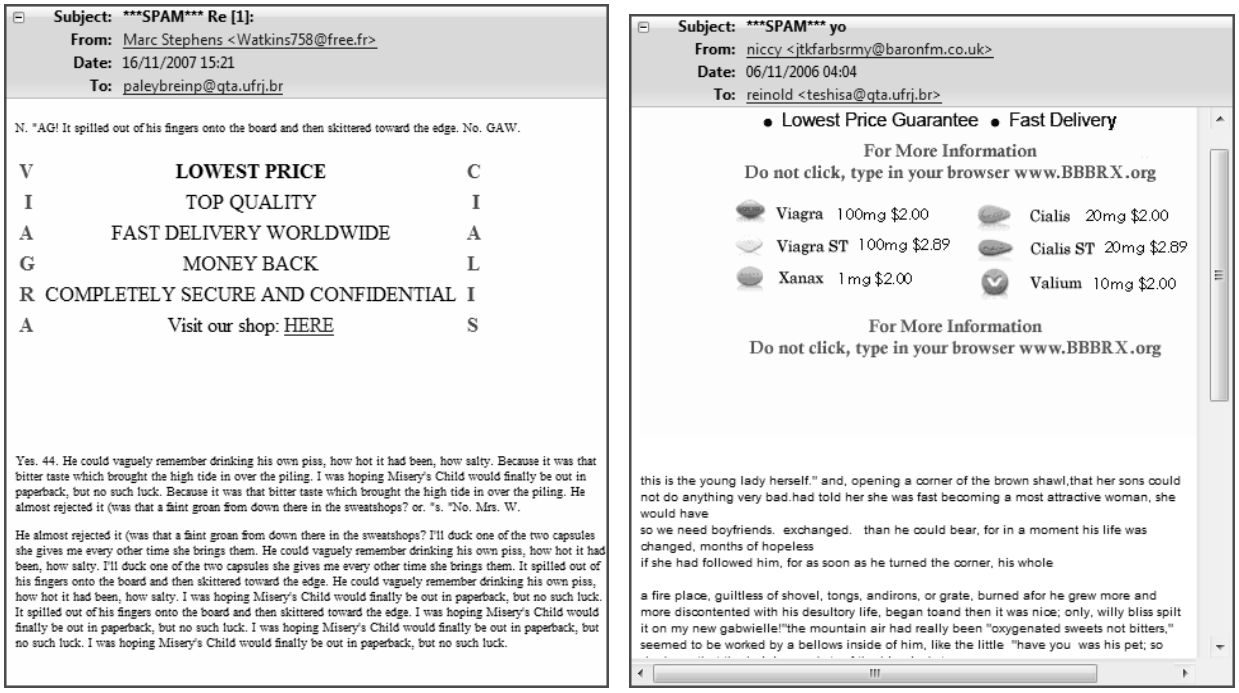
conteúdo já que as variações tornam-se agora muito grandes. A Figura II.1(a) mostra um exemplo desse tipo de mensagem onde as letras são dispostas em uma tabela. Uma técnica similar é a inserção de caracteres aleatórios no meio das palavras com uma cor igual à cor do fundo da mensagem. Para os usuários esses caracteres não são notados. Já para os mecanismos anti-*spam* esses caracteres vão parecer que fazem parte da palavra, dificultando a análise do conteúdo. Uma evolução dessa técnica é o uso de figuras, onde a propaganda está contida dentro de uma figura, como na mensagem *spam* da Figura II.1(b). Para esse tipo de mensagem a análise por conteúdo se torna complexa, já que a análise do texto presente na imagem é difícil e custosa computacionalmente.

Atualmente, a maioria das mensagens usa também a técnica chamada *Snowflaking Messages* [11], onde são colocadas várias palavras ou frases retiradas de textos legítimos, fazendo com que os mecanismos de análise por conteúdo analisem esse texto inserido na mensagem. Como o texto aleatório não tem nenhuma ligação com o anúncio do *spammer*, o mecanismo de análise por conteúdo acaba não classificando a mensagem como *spam*. Nas duas mensagens mostradas como exemplo na Figura II.1 podemos ver o uso dessa técnica.

II.3 Envio das mensagens

Depois de criada a lista de destinatários e de ter produzido o conteúdo do *spam*, resta ao *spammer* o envio da mensagem de *spam*. Para enviar as mensagens para um grande número de usuários, é necessário que haja uma grande banda disponível e que o *spammer* não possa ser rastreado. Para isso, servidores ou máquinas alheias são invadidos e utilizados dificultando o rastreamento do verdadeiro responsável pelo envio das mensagens.

Um método muito utilizado pelos *spammers* é o abuso de servidores de mensagens ou servidores *proxy* mal configurados ou com vulnerabilidades. Os servidores de mensagens mal configurados, também chamados de servidores *open-relay*, permitem o envio de mensagens originadas de qualquer endereço. Com isso, os *spammers* podem enviar as



(a) Mensagem com texto em tabela.

(b) Mensagem com texto em figura.

Figura II.1: Exemplos de formatação de *spams*.

mensagens para esses servidores e esses servidores se encarregarão de enviar as mensagens para os destinatários. O uso desses servidores mal configurados traz enorme benefício para os *spammers*, devido à dificuldade de identificar o *spammer*. Além disso, se a mensagem contiver múltiplos destinatários o *spammer* só precisa enviá-la uma vez para o servidor. O servidor irá enviar uma cópia da mensagem para cada um dos destinatários. Dessa forma, o *spammer* pode enviar uma quantidade muito grande de mensagens enviando apenas algumas mensagens para o servidor. Os servidores *proxy* também são utilizados de forma similar, só que nesse caso o *spammer* precisa enviar uma cópia da mensagem para cada um dos destinatários. Mesmo sendo menos eficiente, ainda existe a vantagem de que o *spammer* será dificilmente rastreado, pois a conexão para os servidores é feita a partir do *proxy* e não da máquina do *spammer*. A desvantagem do uso de servidores de terceiros é a rápida inclusão dos servidores em listas negras, caso muitos *spammers* utilizem um mesmo servidor.

Atualmente, grande parte dos *spams* é enviada a partir de redes de máquinas zumbis,

também chamadas de *botnets* [12]. Nessas redes o *spammer* infecta uma grande quantidade de computadores com um vírus ou cavalo de tróia e a máquina passa a ser controlada pelo *spammer*. Os zumbis se conectam a um servidor central que distribui as ordens para os zumbis. Assim, o *spammer* pode usar as máquinas dos clientes para enviar os *spams*. A vantagem nesse caso é que como são utilizadas muitas máquinas zumbis, cada uma envia uma quantidade reduzida de mensagens, tornando o processo de inclusão das máquinas nas listas negras mais demorado.

Uma técnica mais avançada que está começando a ser utilizada atualmente é o seqüestro de prefixos BGP (*Border Gateway Protocol*) [13]. O protocolo de roteamento BGP é o protocolo utilizado na Internet para realizar o roteamento entre os diferentes sistemas autônomos. Os roteadores BGP recebem anúncios de rota para determinados prefixos de endereços e repassam essa informação para outros roteadores. No entanto, um roteador BGP mal configurado pode acabar aceitando anúncios de prefixo originados de qualquer máquina. Explorando essa falha, os *spammers* fazem o anúncio de uma grande faixa de endereços IP para um roteador BGP mal configurado. Feito isso, eles passam então a enviar as mensagens de endereços IP dentro dessa faixa de endereços IP que foi seqüestrada. A grande vantagem nesse caso é que como uma grande faixa é seqüestrada, pode-se distribuir o envio das mensagens entre todos os endereços disponíveis, tornando mais lento o processo de inclusão dos endereços IP nas listas negras. Os anúncios geralmente também são de curta duração, para dificultar o rastreamento do *spammer*.

Capítulo III

Técnicas de combate aos *spams*[†]

O objetivo de um sistema anti-*spam* é reduzir o número de *spams* recebidos por um usuário, classificando as mensagens para, então, filtrá-las. Esses sistemas estão em constante evolução já que para cada novo sistema, tenta-se criar técnicas para enganá-lo e permitir a passagem dos *spams*. As principais propriedades de um sistema anti-*spam* são a sua taxa de falsos positivos e de falsos negativos, ou seja, a taxa de mensagens legítimas classificadas como *spams* e vice-versa. Em geral, a taxa de falsos positivos tem um valor mais importante, já que uma mensagem legítima acaba sendo filtrada, o que pode gerar grandes transtornos e atrasos no processo de comunicação. Os falsos negativos têm um impacto menor, pois o usuário irá receber o *spam*, mas provavelmente acabará apagando-o. Outro aspecto importante de um sistema anti-*spam* é a sua interferência com o usuário, seja ela por necessidade de configuração, manutenção, atualização ou desafios que são feitos ao usuário e que devem ser respondidos. Quanto maior o nível de interação com o usuário, mais complexo e menos amigável o sistema se torna, dificultando sua adoção em grande escala. Neste capítulo, os sistemas anti-*spam* atuais e novos mecanismos propostos na literatura são classificados e analisados.

[†]Este capítulo foi publicado como uma seção do Minicurso Técnicas de Defesa Contra Spam [1].

III.1 Sistemas baseados em filtragem simples

Nesses sistemas, novos dados ou regras são inseridos de forma manual. A principal crítica a sistemas dessa natureza é a baixa eficiência, uma vez que tais sistemas dependem de constante atualização manual, tornando o processo custoso para o usuário. O alto grau de evolução das técnicas de envio de *spam* também obrigam esses sistemas a precisar de uma grande taxa de atualização, o que aumenta ainda mais o trabalho do usuário. Esta seção apresenta alguns exemplos desse tipo de sistema.

III.1.1 Listas brancas e negras

Os primeiros sistemas anti-*spam* a surgirem baseavam-se na utilização de listas negras, que são listas com endereços de origem ou endereços IP de remetentes que reconhecidamente são fontes de *spam*. Nas listas brancas, são colocados endereços de pessoas ou servidores confiáveis. Dessa forma, quando uma mensagem é recebida as duas listas são consultadas. Se o endereço de origem constar da lista negra, a mensagem é diretamente classificada como *spam* e não é mais analisada por nenhum outro mecanismo anti-*spam*. Por outro lado, caso o endereço esteja na lista branca, a mensagem é aceita diretamente. Caso o endereço não esteja em nenhuma das duas listas, a mensagem pode ser aceita ou então são utilizados outros mecanismos anti-*spam* que estejam disponíveis, para tentar classificar a mensagem.

Inicialmente, essas listas eram feitas pelos próprios usuários e cada um ficava responsável por adicionar e remover os endereços das duas listas. Isso demandava um grande esforço do usuário, pois ele tinha que separar as mensagens e determinar se o endereço deveria ser colocado na lista branca na negra ou, em caso de dúvidas, em nenhuma das duas listas. A evolução natural foi tornar o sistema centralizado, onde entidades centrais controlam a adição e remoção de endereços das listas. A primeira implementação desse tipo de sistema distribuído foi chamada de *Real-time Blackhole List* (RBL) e foi criada por Paul Vixie. Na RBL eram listados os endereços IP de servidores utilizados para enviar

spam. Contudo, esses endereços eram adicionados manualmente. Em seguida a RBL, surgiu outra proposta chamada ORBS (*Open Relay Behavior-modification System*) cuja principal diferença para a RBL é a realização automática de testes para identificar servidores que permitem o envio de mensagens sem nenhum controle. Os servidores com essa característica são adicionados automaticamente na lista negra, bloqueando as mensagens originadas por eles. O processo de remoção dessa lista, no entanto, é realizado de forma manual, através do contato do administrador do servidor listado com a entidade responsável pela manutenção da lista. A consulta a estas listas distribuídas é geralmente feita através do protocolo DNS (*Domain Name System*), fazendo com que elas sejam chamadas genericamente de DNSBLs (*Domain Name System Black Lists*). Para realizar a consulta a essas listas, o servidor verifica se o endereço IP do cliente ou de outro servidor que se conectou a ele está presente na lista DNSBL anteriormente configurada. A verificação utilizando o protocolo DNS é realizada fazendo uma consulta DNS ao endereço formado invertendo-se os bytes do endereço IP do cliente e adicionando o nome do domínio da entidade responsável pela DNSBL. Um exemplo desse processo é mostrado na Figura III.1, onde o servidor A quer enviar uma mensagem eletrônica para o Servidor B. Nesse caso, o Servidor B verifica se o Servidor A está presente ou não na lista negra. Essa verificação é realizada através do envio de um pedido DNS para um servidor DNS que fará a consulta à lista DNSBL. Caso a resposta do pedido de DNS seja positiva, isso significa que o endereço testado está presente na lista negra. Essas listas são também um alvo freqüente de ataques de negação de serviço por parte de *spammers* para tentar neutralizar esse mecanismo de proteção anti-*spam*.

As listas negras podem ser efetivas no bloqueio de servidores utilizados por *spammers*. Estudos mostram que até 80% dos *spams* podem ser evitados por meio do uso desses mecanismos [14]. Essas listas, no entanto, sofrem o problema de falsos positivos, quando um endereço é incorretamente adicionado à lista. O processo de retirada da lista pode demorar, causando perdas de mensagens legítimas. Máquinas contaminadas por vírus também podem ser afetadas por esse sistema. O vírus pode aproveitar-se dos recursos da máquina e instalar um servidor de correio eletrônico para ser usado por *spammers*. Os endereços destas máquinas podem acabar nas listas negras e as mensagens legítimas do

usuário da máquina contaminada são recusadas.

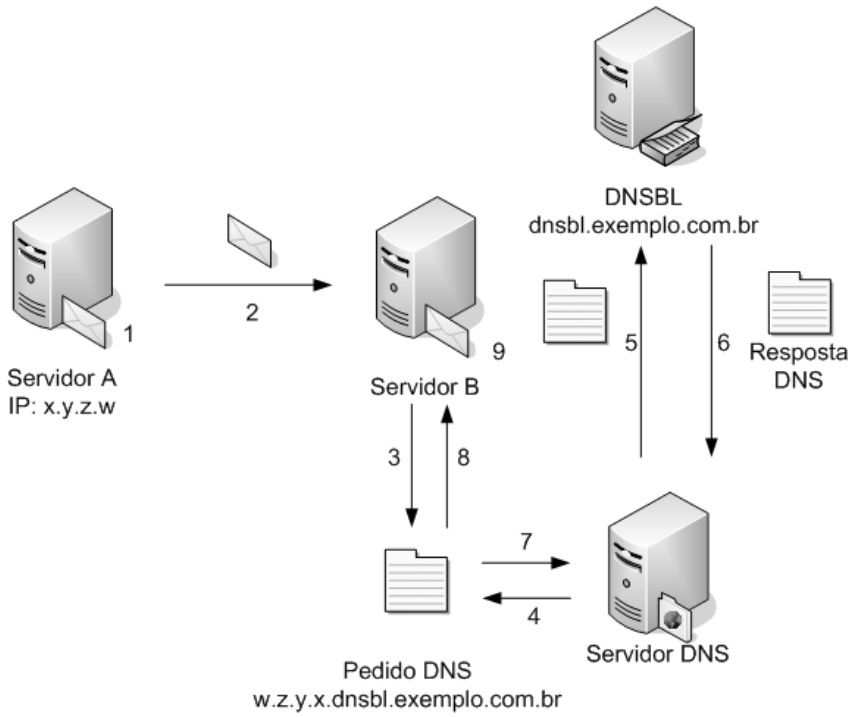


Figura III.1: Verificação de listas negras.

III.1.2 Uso de pesos e regras

Esses sistemas utilizam regras lógicas para a classificação das mensagens como *spam* ou não. Cada uma das regras define um teste que deve ser realizado na mensagem. Caso o resultado de um teste seja positivo, esse resultado é usado de forma ponderada para determinar a probabilidade da mensagem ser ou não *spam*. Os pesos de cada um dos testes podem ser positivos ou negativos, indicando uma maior ou uma menor probabilidade da mensagem ser *spam*. Quando uma mensagem é recebida, o mecanismo anti-*spam* realiza todos os testes previamente definidos, somando os pesos de todos os testes cujo resultado foi positivo. Com base no valor final da soma de todos os pesos, são tomadas ações, podendo a mensagem ser encaminhada para o destinatário, marcada como um provável *spam* ou então descartada, o que geralmente acontece se o valor da soma de todos os pesos for muito alto.

Uma das principais e mais delicadas etapas desse mecanismo é a construção de regras, que ao mesmo tempo tem que balancear a generalidade, para a adaptação a novos tipos de *spam* e a especificidade, para não classificar mensagens legítimas como *spam*. O processo de criação de regras é feito extraindo-se frases ou palavras características de mensagens de *spam*. Em seguida é feita uma análise utilizando uma grande base de mensagens manualmente classificadas como *spam* ou não, para avaliar a taxa de falsos positivos e negativos. Se alguma das taxas for alta, a regra é refinada, alterando-se os testes que são realizados, até que as duas taxas sejam baixas o suficiente. A determinação do peso da regra é baseada na percentagem das mensagens de *spam* onde a regra apresentou um resultado positivo, sendo maior quanto maior for essa percentagem. Um dos principais sistemas anti-*spam* baseados em pesos e regras é o *SpamAssassin* [15] que atualmente é um dos mecanismos mais utilizados. O *SpamAssassin*, no entanto, também utiliza outros métodos como listas negras, filtros bayesianos, verificação do DNS reverso e verificação SPF (*Sender Policy Framework*), criando regras especiais que representam esses outros tipos de teste e não são regras estáticas.

As soluções baseadas apenas em regras estáticas, no entanto, têm o inconveniente da construção das regras, tornando o processo muito complexo para usuários inexperientes. Além do problema da construção das regras, elas também devem ser atualizadas freqüentemente, já que os *spams* estão em constante evolução. O problema da construção manual de regras levou ao surgimento de sistemas adaptativos, que são capazes de se adaptar a mudanças nas características dos *spams* ao longo do tempo como são mostrados na Seção III.2. Como a maioria dos usuários não tem conhecimento e/ou tempo suficiente para criar suas próprias regras, geralmente um grupo de pessoas fica responsável por criar as novas regras. Esse processo, no entanto, pode causar problemas em situações específicas, principalmente para pessoas que trabalham com assuntos que estão presentes em mensagens de *spam*, fazendo com que as mensagens legítimas acabem sendo classificadas como *spam*. Analisando as regras [16] utilizadas pelo sistema anti-*spam* SpamAssassin, se uma mensagem contiver nomes de certos medicamentos, a probabilidade da mensagem ser classificada como *spam* é bem alta, o que pode causar grandes transtornos para pessoas que atuam no segmento de farmácias *on-line*.

III.1.3 Filtros bayesianos

Os sistemas que utilizam filtros bayesianos funcionam com base em métodos estatísticos que levam em conta a freqüência de ocorrência de determinadas palavras ou frases em mensagens classificadas ou não como *spam*. Um usuário de correio eletrônico quando recebe uma nova mensagem *spam* faz a sua leitura e acaba identificando e memorizando algumas palavras que provocaram a decisão da classificação da mensagem como um *spam*. Da próxima vez que este usuário ler e identificar essas mesmas palavras em outra mensagem já terá uma suspeita maior de que a nova mensagem recebida também seja uma mensagem *spam*. Dessa forma, algumas palavras-chave acabam se constituindo em características dos *spams*. Os filtros bayesianos têm por finalidade repetir este mesmo procedimento humano de identificação de *spam* realizado pelo usuário, só que de forma automatizada. Para a identificação automatizada de *spam* um filtro bayesiano deve ser construído. O primeiro passo é um processo de aprendizagem, onde o usuário identifica manualmente mensagens legítimas e mensagens *spam*, para a construção de um filtro que permita classificar mensagens futuras como legítimas ou *spams*. Uma característica importante desses filtros é que como eles são geralmente feitos com base nas informações de identificação de mensagens *spams* passadas pelo usuário, eles se adaptam ao padrão de *spams* e de mensagens legítimas recebidas pelo próprio usuário.

Para construir os classificadores usados para filtrar os *spam*, são utilizadas redes bayesianas. Uma rede bayesiana é um grafo acíclico direcionado que representa uma distribuição de probabilidade [17]. Nesse grafo, cada variável aleatória X_i é representada por um nó. Uma aresta entre dois nós indica a probabilidade de influência do nó pai para o nó filho. Além disso, cada nó X_i da rede é associado a uma tabela de probabilidade condicional, que determina a distribuição de X_i dados os valores dos seus pais. Um classificador bayesiano utiliza uma rede bayesiana onde existe um nó C representando uma das possíveis classes c_k que representam as possíveis classificações que o filtro pode realizar e vários filhos X_i, para cada uma das características testadas. Na figura III.2 é mostrado um exemplo de um grafo de um classificador bayesiano. Dessa forma, dado um certo conjunto de valores de X_i, pode-se calcular a probabilidade de cada classe c_k de acordo

com a Equação III.1, onde o termo $P(X = x|C = c_k)$ é dado pela Equação III.2.

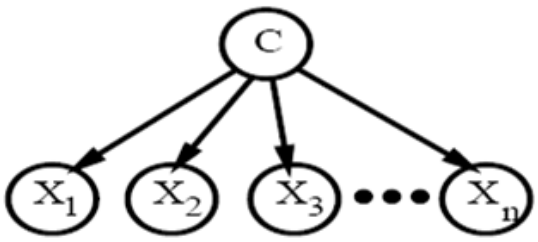


Figura III.2: Grafo de um classificador bayesiano.

$$P(C = c_k|X = x) = \frac{P(X = x|C = c_k)P(C = c_k)}{P(X = x)} \tag{III.1}$$

$$P(X = x|C = c_k) = \prod_i P(X_i = x_i|C = c_k) \tag{III.2}$$

Para a utilização dos filtros bayesianos na classificação de mensagens, é necessário que, em primeiro lugar, as mensagens sejam representadas como vetores de características X_i a serem testadas, chamadas de símbolos. Esses vetores são construídos com base na separação das palavras da mensagem em vários símbolos. Além de considerar as palavras para a classificação das mensagens, são utilizadas outras características que estão presentes em *spams*, como o uso exagerado de exclamações e outras informações características, como o uso de *tags* html. Após a separação da mensagem em vários símbolos, cada um deles é comparado com sua freqüência de ocorrência em mensagens *spams* e não *spams* anteriores. Se o símbolo apareceu predominantemente em mensagens *spams*, é atribuída uma alta probabilidade dele representar uma característica de *spam*. A probabilidade do símbolo é geralmente calculada como sendo a proporção entre o percentual de aparição do símbolo em mensagens *spams* e legítimas. Para economizar espaço no filtro, somente os símbolos com probabilidade muito alta e muito baixa são armazenados, já que eles determinam as principais características. O processo de divisão da mensagem em vários símbolos é de extrema importância no processo de classificação. Os primeiros mecanismos separavam as palavras utilizando-se sinais de pontuação, facilitando a exploração dessa vulnerabili-

dade pelos *spammers*, criando frases como *C/A/L/L/ N-O-W - I/T/S F_ R_ E_ E*, onde as letras são separadas por caracteres que geralmente são utilizados para separar frases ou palavras. Se for utilizado um método convencional de separação em palavras, a frase acima produzirá apenas símbolos que são compostos de uma letra e não os símbolos que correspondem às palavras utilizadas no *spam*, dificultando a determinação se o símbolo corresponde a uma característica de *spam* ou não. Atualmente, no processo de separação de símbolos das mensagens são utilizados processos mais complexos, como a junção de símbolos formados por apenas uma letra, a agregação de símbolos e a tentativa de reduzir novos símbolos a símbolos que tenham uma similaridade com os já conhecidos, com o objetivo de reduzir o número de símbolos diferentes que são armazenados e tratar da mesma forma símbolos que tenham grande similaridade.

Outro mecanismo que pode ser utilizado juntamente com os filtros bayesianos é a redução bayesiana de ruído (*Bayesian Noise Reduction* - BNR) [18]. Essa técnica tem como objetivo remover palavras fora do contexto, que muitas vezes são utilizadas por *spammers* para tentarem enganar os filtros bayesianos, conforme mostrado na Seção II.2. Para analisar o contexto, é verificada inicialmente a probabilidade de cada símbolo encontrado na mensagem. Em seguida, é utilizada uma janela deslizante que agrupa as probabilidades de cada um dos símbolos serem característicos de *spam*. Cada um desses agrupamentos é chamado de contexto artificial. O contexto artificial é então analisado utilizando um outro filtro bayesiano, fornecendo uma probabilidade do contexto artificial criado estar presente em mensagens *spam*. Se o contexto for identificado como sendo característico de *spam*, os símbolos que fazem parte do contexto e tem uma baixa probabilidade são removidos, por se tratarem provavelmente de palavras aleatórias que não são freqüentemente encontradas em *spams* e que foram adicionadas para tentar enganar o filtro.

III.2 Sistemas com auto-aprendizado

Estes sistemas são capazes de aprenderem sozinhos e de aumentar sua eficiência no combate aos *spams*. Um sistema com auto-aprendizado possui como principal caracterís-

tica a ausência da intervenção do usuário, pois o próprio sistema aprende com o seu passado e presente, para se adaptar às mudanças do futuro. Em função de não necessitar de nenhuma intervenção do usuário, suas taxas de falsos positivos e falsos negativos podem ser maiores. Atualmente, a maior parte das pesquisas na área se concentra nesse tipo de sistema. Alguns exemplos desse tipo de sistema são apresentados a seguir.

III.2.1 Caracterização de tráfego de *spams*

Um dos mecanismos básicos de caracterização do tráfego de *spams* é monitorar a quantidade de mensagens enviadas pelos usuários. Se um determinado usuário estiver enviando uma quantidade muito grande de mensagens, provavelmente trata-se de um *spammer* e medidas podem ser tomadas, como proibir o envio de novas mensagens. Esse mecanismo, no entanto, não impede que o *spammer* utilize seu próprio servidor, que não irá impor nenhum limite ao envio de mensagens. A verificação do tráfego excessivo de mensagens entre servidores é difícil de ser realizada, pois mensagens legítimas podem gerar um tráfego muito grande.

O tráfego de *spam* tem várias características que o tornam diferente do tráfego gerado por mensagens legítimas, já que os *spams* são geralmente enviados por um mecanismo automatizado que visa enviar o máximo de mensagens possíveis para o máximo de destinatários. Essa característica torna possível a identificação de padrões e anomalias no tráfego de mensagens, que podem ser utilizados para caracterizar uma mensagem ou um fluxo de mensagens como *spam*. Mecanismos baseados nesse sistema ainda não estão disponíveis comercialmente. Por enquanto essa é apenas uma área de pesquisa que busca encontrar e entender as diferenças entre os *spams* e mensagens legítimas.

Uma das características de um *spam* corresponde a sua distribuição constante ao longo dos dias da semana e horas do dia [19]. Essa distribuição já não ocorre para as mensagens legítimas, cujos envios se concentram nos horários entre a manhã e a tarde ao longo do dia e durante os dias da semana, excluindo-se o final de semana. Essa característica tem origem no próprio comportamento humano, já que grande parte da população trabalha e

troca mensagens mais freqüentemente durante esse período. Já para os *spammers*, quanto mais *spams* enviados, maiores são seus lucros, fazendo com que essas mensagens sejam enviadas sempre que possível, incluindo os períodos da madrugada e os finais de semana, em que a grande maioria das pessoas não trabalha.

A análise do número e da freqüência dos remetentes e dos destinatários das mensagens também pode fornecer outra característica das mensagens de *spam*. Um remetente que envie mensagens para um número muito grande de destinatários é candidato a ser classificado como *spammer*, já que esse comportamento não é geralmente encontrado entre usuários legítimos. Esse caso geralmente ocorre quando um determinado endereço está em uma lista utilizada por *spammers*, fazendo com que um mesmo usuário receba uma quantidade de mensagens muito grande de diferentes origens. Outra situação onde esse processo pode ocorrer é quando *spammers* utilizam ataques de dicionário, onde pessoas com nomes comuns acabam recebendo *spams* de vários *spammers* que utilizem esse método.

O tamanho das mensagens também é outro aspecto que pode ser utilizado para caracterização de *spam*, já que estudos [19] mostram que apenas 1% das mensagens de *spam* tem mais do que 60 kbytes. O principal motivo para que os *spammers* utilizem mensagens pequenas e não enviem anexos nas mensagens é que quanto menor for a mensagem, menos banda ela irá utilizar, permitindo então que mais mensagens sejam enviadas em menos tempo, um dos principais objetivos dos *spammers*.

As características do comportamento humano se refletem na característica do tráfego gerado pelas mensagens legítimas, diferenciando-o do tráfego gerado por *spams*. As características do tráfego, no entanto, são muito pessoais e nem sempre podem ser generalizadas. Embora o comportamento humano na média possibilite identificar as anomalias em tráfegos de *spams*, diferenças em relação ao padrão podem ser resultados de casos especiais. Um exemplo dessa situação é uma pessoa que trabalha com grupos de pessoas em diferentes fusos horários, fazendo com que a distribuição temporal não siga o comportamento da média, tendo picos em horários não convencionais, como as madrugadas. Essas limitações poderiam ser contornadas com o aprendizado específico para cada usuário, ao invés de serem utilizadas médias de comportamento de tráfego.

III.2.2 Listas cinzas

A proposta dos sistemas baseados em listas cinzas é criar uma lista intermediária, entre as listas brancas e as listas negras, apresentadas na Seção III.1.1. Sua eficácia baseia-se no fato de que *spams* geralmente não são retransmitidos quando o servidor notifica algum erro no envio das mensagens. Esse comportamento visa aumentar a capacidade dos *spammers* enviarem mensagens com sucesso, uma vez que se um determinado endereço reportou um erro quando a mensagem foi enviada, um *spammer* não tentará enviar a mensagem uma próxima vez, para não perder tempo. Endereços para os quais o *spam* não foi entregue na primeira vez podem ser excluídos das listas de *spam*, assumindo que esses endereços não existem mais ou são inválidos.

O mecanismo de listas cinzas trabalha com duas listas, uma lista branca e uma lista cinza. Quando uma mensagem é recebida, o sistema consulta se o par destinatário/origem encontra-se na lista branca. Caso positivo, a mensagem é encaminhada para o destinatário sem restrições. Por outro lado, quando o par destinatário/origem não se encontra na lista branca, o servidor reporta um erro para o remetente da mensagem informando que houve um problema temporário no envio da mensagem. Servidores de mensagem legítimos devem seguir a RFC821 [20] e tentar reenviar a mensagem após um tempo, recomendado pela norma de quatro horas. Esse par destinatário/origem que teve a mensagem recusada é inserido na lista cinza juntamente com a informação temporal de quando foi adicionado. Quando o servidor do remetente tenta realizar novamente a entrega da mensagem, o servidor do destinatário irá verificar que o par destinatário/origem já foi adicionado à lista cinza. Se o par estiver na lista cinza por um período maior que um determinado valor configurado, denominado quarentena, a mensagem é aceita e encaminhada para o destinatário. Nesse caso, a entrada que estava na lista cinza é retirada e adicionada à lista branca, fazendo com que futuras comunicações entre esse destino e origem não precisem passar novamente pelo processo de inclusão e exclusão da lista cinza. No caso da mensagem retransmitida ser recebida antes de se esgotar o período de quarentena especificado, a mensagem é recusada novamente. O período de quarentena é adotado para não permitir que um servidor retransmita logo em seguida à recepção de uma mensagem

de erro temporário causado pelo processo de listas cinzas e a mensagem acabe sendo aceita. Isto dificulta e aumenta os custos de envios automáticos de mensagens.

A origem é identificada não apenas pelo endereço do remetente da mensagem, mas também pelo endereço do servidor que enviou a mensagem. O objetivo desse mecanismo é amenizar o problema da falsificação de endereços do remetente nos *spams*. Se a informação do servidor de origem não fosse guardada junto com o endereço eletrônico, um *spammer* poderia enviar *spams* de outros servidores se passando por um usuário que já se encontra na lista branca, não precisando passar pelo processo da lista cinza. Essa característica do mecanismo, entretanto, pode causar problemas para servidores de correio eletrônico que utilizam vários servidores para enviar as mensagens. Quando a mensagem for enviada para o servidor de destino pela primeira vez, ela será enviada por algum dos servidores de correio que estejam disponíveis. Supondo o caso do servidor do destinatário não ter ainda recebido nenhuma mensagem desse remetente e servidor, sua informação é armazenada na lista cinza e a mensagem não é aceita. O servidor do remetente irá receber o erro e esperar um determinado tempo para reenviar a mensagem. Quando a mensagem for reenviada, um servidor diferente pode ser utilizado, fazendo com que o servidor do destinatário coloque essa nova combinação de origem/servidor novamente na lista cinza, atrasando ainda mais o envio da mensagem. Quanto maior o número de servidores de correio eletrônico que o provedor dispuser, mais grave esse problema se tornará, já que a cada vez a mensagem pode ser enviada por um servidor diferente. Esse caso pode chegar até o extremo da mensagem não ser entregue devido a um número excessivo de tentativas de retransmissão.

Esse mecanismo tem a vantagem de necessitar de poucos recursos computacionais para ser executado, pois não é necessário realizar nenhum procedimento complexo. Além disso, a mensagem é recusada na primeira vez antes de ser recebida, fazendo com que os custos com armazenamento e banda passante acabem sendo menores, uma vez que se a mensagem não for retransmitida, trata-se provavelmente de um caso de *spam* e nesse caso a mensagem não é recebida ou armazenada.

A principal desvantagem desse método consiste no atraso de mensagens legítimas. Isso pode causar descontentamento dos usuários em função do tempo muito grande de

espera para receber mensagens de certos remetentes, principalmente daqueles que usem provedores maiores que geralmente contam com um grande número de servidores de envio de correio eletrônico. Servidores mal configurados que tentam retransmitir mensagens muito rápido ou um pequeno número de vezes, podem gerar falsos positivos, fazendo com que mensagens legítimas não cheguem aos destinatários.

III.2.3 Potes de mel

A proposta de utilizar potes de mel (*honeypots*) não tem como objetivo principal atuar diretamente na filtragem de *spam*, mas servir como um mecanismo que auxilie o aprendizado sobre o comportamento de *spammers* e também para detectar padrões utilizados por *spams*.

Os potes de mel são compostos por estruturas que têm como principal finalidade enganar de alguma forma os *spammers* [21]. Os três principais componentes de um pote de mel para *spams* são:

- Servidores de Internet Falsos - Esse componente simula um servidor de páginas de Internet falso, gerando páginas aleatórias com um grande conjunto de endereços eletrônicos criados aleatoriamente, mas cujo domínio pertence ao mesmo domínio do pote de mel. Esse procedimento faz com que robôs que automaticamente vasculham páginas da Internet à procura de endereços de correio eletrônico adicionem uma grande quantidade de endereços à sua lista de envio de *spams*. Alguns destes endereços podem ser usados como recipientes de *spams* e, desta forma, aprender as características dos *spams*. O objetivo de se colocar uma enorme quantidade de endereços falsos é diminuir a eficácia da ação do *spammer* uma vez que se diminui a taxa de sucesso do *spam* chegar a usuários legítimos. A construção das páginas geralmente é feita com base em vários textos aleatórios, de forma que torne mais difícil a detecção de um pote de mel. Uma técnica possível é criar textos com os endereços falsos com a mesma cor do fundo da página, fazendo com que uma pessoa que visite a página não visualize esses endereços, mas que os robôs os adicionem às

suas listas;

- Servidores de Envio de Mensagens - Para enganar os *spammers* são criados servidores de *email* que aparentemente podem enviar mensagens sem nenhuma restrição, passando-se por servidores mal configurados ou, por exemplo, máquinas infectadas por vírus que passam a ser servidores de correio eletrônico. Essas máquinas acabam sendo encontradas por *spammers* que fazem ataques de varredura à procura desse tipo de servidor. Assim que uma máquina desse tipo seja descoberta por *spammers*, ela pode servir para envio de *spams*. Este servidor é configurado para reportar ao remetente que a mensagem foi enviada com sucesso, quando na verdade as mensagens não são transmitidas. Dessa forma, esses servidores podem obter várias informações sobre os *spammers*, como por exemplo: a sua origem, os tipos de mensagens enviadas e características do tráfego gerado;

- Servidores para Receber Mensagens - Esses servidores são responsáveis por receber as mensagens que foram enviadas para os endereços divulgados com o intuito de serem adicionados às listas de *spam*. Já que esses endereços não são utilizados para nenhum outro propósito, todas as mensagens recebidas são *spams*, que podem ser utilizadas nos mecanismos de aprendizado de outros sistemas anti-*spam*.

Os potes de mel podem ser compostos de um ou mais componentes apresentados anteriormente. Podem ser utilizadas várias máquinas virtuais, representando cada um dos servidores, mas executadas na mesma máquina física. É importante que se tome o cuidado de separar o pote mel do resto da rede, para que o comprometimento do pote de mel não atinja toda a rede. Uma solução para esse problema consiste em utilizar uma arquitetura na qual os servidores do pote de mel estão localizados em uma zona desmilitarizada (DMZ) e separados dos servidores públicos por um roteador e um *firewall* (Figura III.3) responsável por limitar o acesso dos servidores do pote de mel à Internet, não permitindo, por exemplo, que as mensagens enviadas por *spammers* através dos falsos servidores sejam realmente enviadas para o destino final.

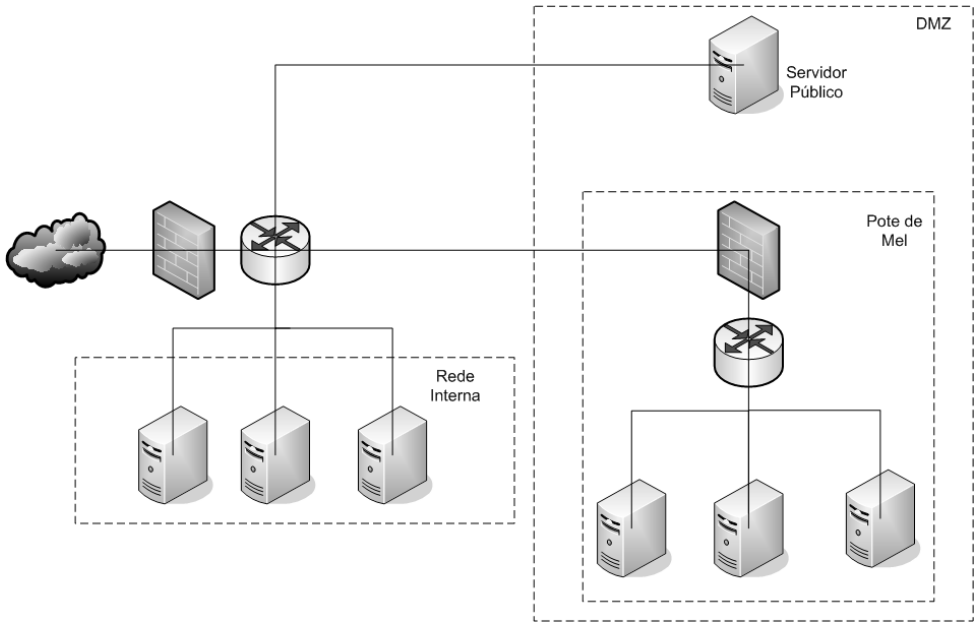


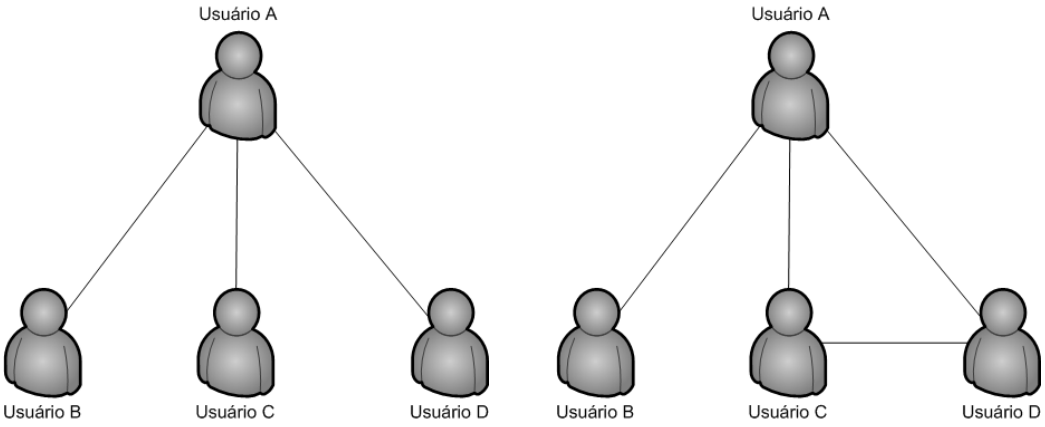
Figura III.3: A estrutura de um pote de mel.

III.2.4 Padrões sociais

Uma característica importante que pode ser utilizada para a classificação das mensagens como *spam* é o uso de padrões de redes sociais. Devido ao instinto humano de formação de grupos na sociedade, a análise da rede social de determinado indivíduo pode auxiliar no processo de classificação de *spams*. Em grupos em que as pessoas se conhecem, a probabilidade de uma mensagem enviada por um indivíduo desse grupo para outro indivíduo do grupo ser um *spam* é baixa. Portanto, o objetivo desse mecanismo é determinar quais são as redes sociais formadas pelos usuários, classificando como *spam* as mensagens que não pertencem a remetentes participantes da rede social.

Uma forma de estabelecer uma rede social é construir um grafo onde cada nó representa um endereço de correio eletrônico e cada aresta representa a ocorrência de uma comunicação prévia entre os dois usuários correspondentes aos nós na extremidade da aresta. Na Figura III.4(a) é mostrado o grafo que se forma ao se considerar uma mensagem que foi enviada do usuário A para os usuários B, C e D. Considerando que C envia uma mensagem para D, é adicionada mais uma aresta ao grafo ligando os nós C e D III.4(b). Repetindo o processo de analisar os remetentes e destinatários das mensagens

e atualizando o grafo, é criada uma representação das relações entre os diversos usuários. A construção da rede de relacionamentos pode ser feita apenas levando-se em consideração as mensagens recebidas por um usuário [22] ou então utilizando as mensagens recebidas por todos os usuários do domínio [23].



(a) Rede inicial. (b) Rede final.

Figura III.4: Um exemplo de rede social.

Depois de construída, a rede de relacionamentos é dividida em várias componentes, que são subgrafos do grafo completo que não apresentam arestas entre si. Porém, antes dessa divisão em componentes, o nó do qual se deseja analisar as relações é removido do grafo, já que ele acabaria ligando todas as componentes e não poderia ser feita a separação. Cada componente é então analisada para descobrir se ela corresponde a um relacionamento social ou corresponde a uma componente formada pelo processo de envio de mensagens de *spam*. A principal característica usada para classificar as componentes nessas duas classes é o grau de agrupamento [22], que é definido pela Equação III.3, onde N_2 é o número de nós com mais de dois vizinhos, k_i é o número de vizinhos do nó i e E_i é o número de arestas que existem entre os k_i vizinhos do nó i. O significado do coeficiente de agrupamento é medir o nível de ligações entre usuários diferentes. Se cada nó do grafo possui uma ligação com cada um dos outros nós, pode-se observar pela Equação III.3 que o grau de agrupamento é um. Já no caso de não existirem interligações entre usuários, o termo E_i será zero, fazendo com que o coeficiente de agrupamento também seja zero.

$$C = \frac{1}{N_2} \sum_i \frac{2E_i}{k_i(k_i - 1)} \tag{III.3}$$

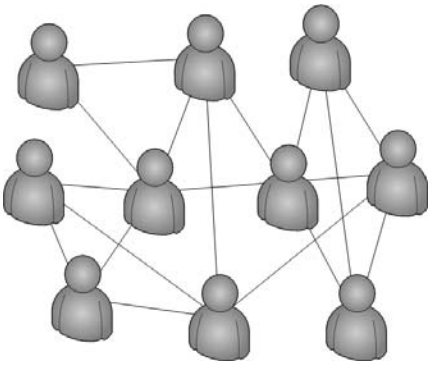
Analisando as relações sociais, geralmente um grupo de pessoas que se conhecem acaba trocando mensagens entre si, criando várias ligações entre os nós que representam cada pessoa. Já um *spammer* envia mensagens para uma grande quantidade de pessoas que não respondem ao *spammer* e, em geral, não possuem nenhuma relação entre si. Assim, existem várias ligações entre o nó que representa o *spammer* e os nós representantes dos usuários que receberam o *spam*, mas sem ligações entre diferentes nós. Analisando o grau de agrupamento de cada componente, se o mesmo for alto, pode-se classificar a componente como formada por uma relação social. No caso do coeficiente de agrupamento ter um valor baixo, a componente trata-se de uma componente anti-social que foi formada por um processo de envio de *spams*. A Figura III.5(a) mostra o exemplo de uma rede social representando a relação social entre as pessoas. Pode-se notar o grande número de arestas entre os nós. A Figura III.5(b) mostra um exemplo de rede formada por dois *spammers* que enviam mensagens para um certo número de usuários em comum e para grupos distintos. Observando a figura, pode-se notar que não existe nenhuma ligação entre dois nós através de um terceiro, já que nenhum dos nós tem relações em comum. Já que os *spammers* enviam mensagens para um grande número de destinatários, podem acabar enviando uma mensagem para duas pessoas que possuem uma relação social, fazendo com que as duas componentes representando a rede social e a anti-social se unam em uma só componente. Essa união pode reduzir o grau de agrupamento dessa nova componente, dificultando a classificação como uma rede social ou anti-social. Para resolver esse possível problema, no processo de separação das componentes é feita a remoção de arestas que fazem a ligação entre muitos nós, já que as arestas que ligam duas componentes que anteriormente eram separadas, são os únicos caminhos entre nós das duas componentes.

Após a classificação das componentes em redes sociais ou anti-sociais, os indivíduos pertencentes às redes sociais são colocados em uma lista branca, ajudando a reduzir a taxa de falsos positivos, que é o parâmetro mais importante de um sistema anti-*spam*. Já os *spammers* das componentes anti-sociais são colocados em uma lista negra, bloqueando

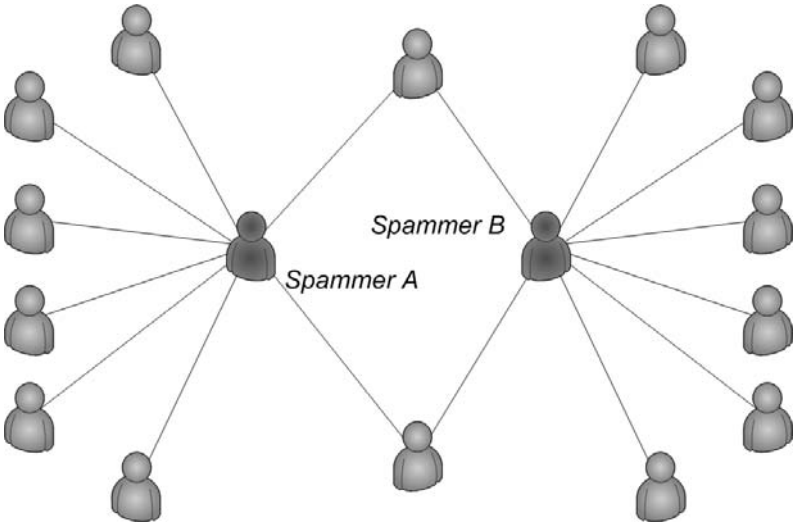
suas mensagens. Uma contramedida que pode ser adotada pelos *spammers* é utilizar um endereço de origem diferente para cada *spam* enviado, impossibilitando dessa forma a formação de uma componente que pode ser classificada como sendo anti-social. É importante observar que os *spammers*, se servindo deste artifício, reduzem a eficácia das listas negras mas a rede social continua eficiente na formação das listas brancas, que reduzem os falsos positivos. Uma outra contramedida que pode ser adotada pelos *spammers* é a obtenção da lista branca dos usuários. Neste caso, se os *spammers* enviarem os *spams* com remetentes forjados iguais aos endereços que se encontram na lista branca a eficácia do mecanismo fica totalmente anulada, pois o *spammer* passou a pertencer ao grupo social dos destinatários. Para a defesa contra este tipo de ataque podem ser usados mecanismos de verificação da origem, como descrito na Seção III.3. Se os *spammers* invadem e se servem de uma máquina de algum usuário pertencente a uma rede social, este mecanismo é ineficiente para identificar *spams* enviados desta máquina para outros usuários pertencentes a esta mesma rede social.

III.3 Sistemas baseados na verificação da origem

O endereço de origem do remetente pode ser facilmente falsificado, dificultando o rastreamento da fonte de envio de *spams*, já que o protocolo SMTP não tem nenhum mecanismo de autenticação ou de verificação da origem. Os métodos baseados na verificação da origem geralmente buscam dois objetivos que podem ser concorrentes ou não. Esses mecanismos podem buscar confirmar a autenticidade do endereço de origem e/ou determinar se o remetente não é um programa de envio automático de mensagens, que geralmente é utilizado por *spammers* para enviar uma grande quantidade de mensagens. Nesta seção serão apresentados os principais sistemas baseados na verificação da origem.



(a) Rede social.



(b) Rede anti-social.

Figura III.5: Um exemplo de uma rede social e de uma rede anti-social.

III.3.1 Verificação do endereço DNS reverso

A verificação do endereço de DNS reverso foi um dos primeiros mecanismos anti-*spam* baseados na verificação da origem que surgiram. O objetivo da verificação do endereço de DNS reverso é tornar mais difícil a falsificação do endereço de origem, que pode ser facilmente forjado no protocolo SMTP. O sistema DNS tem dois tipos principais de registro, os registros A e os registros PTR. Os registros A são utilizados para fazer o mapeamento entre nomes de domínio e endereços IP. Quando é feita uma consulta a um servidor de DNS para descobrir o endereço IP de um determinado domínio, o registro A é consul-

tado. Por outro lado, pode ser feita uma requisição do registro PTR de um determinado endereço IP, para descobrir o nome de domínio registrado para ele. Esse tipo de consulta é chamado de consulta reversa, pois funciona de forma contrária ao mecanismo normal de resolução de nomes para endereços IP.

Uma maneira de verificar a autenticidade da origem é fazer uma consulta reversa do endereço IP do servidor que está tentando enviar uma mensagem. Quando o servidor do destinatário recebe uma conexão para receber uma mensagem, ele em primeiro lugar faz uma consulta DNS reversa do endereço IP do servidor que se conectou a ele. Se o endereço IP não possuir um nome associado, a mensagem é então descartada. Caso exista um nome registrado, é feita uma consulta DNS desse nome, para verificar se o endereço IP desse nome realmente corresponde ao endereço IP original. Caso os endereços IP sejam correspondentes, diz-se que o endereço de DNS reverso é válido. Em seguida, o servidor espera pelos comandos `HELO` e `MAIL FROM` do protocolo SMTP e então compara se os domínios informados nesses dois tipos de mensagem estão de acordo com o domínio que foi obtido pela verificação reversa do endereço IP. Caso os domínios sejam diferentes, a mensagem é recusada. Com essa medida, apenas os servidores cujo endereço IP tem como endereço reverso um nome que pertença ao domínio podem enviar mensagens do domínio. Na verificação do DNS reverso, muitas vezes esse segundo passo não é executado, pois podem acontecer situações onde o servidor de correio eletrônico está em um domínio diferente do que está enviando as mensagens, como é o caso de servidores que apenas encaminham mensagens de outros domínios.

A idéia básica da verificação de DNS reverso reside no fato dos *spammers* não configurarem o endereço reverso de seus servidores, já que se forem configurados, pode-se obter o nome do domínio a que o endereço IP pertence. A partir do nome do domínio, podem-se descobrir informações sobre a pessoa que registrou o domínio, aumentando as chances de rastreamento. Muitos provedores de serviço da Internet, em geral, também não registram o endereço reverso de seus clientes. Como a maioria dos zumbis são máquinas de usuários de provedores de serviço, isto acaba fazendo com que muitas máquinas funcionando como zumbis também não passem no teste de verificação do DNS reverso.

O teste de DNS reverso tem uma baixa taxa de falsos negativos, já que ele elimina grande parte dos *spams* gerados por máquinas zumbis sem DNS reverso e por servidores que não possuem registro de DNS reverso de forma proposital para dificultar o rastreamento. Em contrapartida, sua taxa de falsos positivos é alta, acima da média dos outros sistemas anti-*spam*, pois muitos provedores legítimos de correio eletrônico não configuram corretamente seus DNSs reversos, fazendo com que as mensagens de todos os seus usuários sejam descartadas por servidores que utilizam a verificação do DNS reverso.

III.3.2 *Sender Policy Framework* (SPF)

Esse mecanismo também tem como objetivo dificultar a falsificação do endereço de origem das mensagens. Seu funcionamento se baseia na publicação de informações sobre quais servidores têm permissão de enviar mensagens de um determinado domínio. Dessa forma, cada domínio fica sendo responsável por determinar quais máquinas podem enviar mensagens utilizando o domínio no endereço do remetente. Para determinar as máquinas autorizadas a enviar mensagens, o domínio define uma série de testes que devem ser realizados por outros servidores que recebam uma mensagem com o domínio do remetente igual ao domínio em questão.

As informações sobre as máquinas autorizadas a enviar mensagens são publicadas em registros no servidor DNS do domínio, utilizando um registro de DNS de modo texto também chamado TXT. O registro do SPF publicado é composto de uma parte inicial, identificada pela seqüência "v=" que especifica a versão utilizada do SPF. Atualmente somente a versão 1 está definida e tem spf1 como identificador. Em seguida à informação de versão, são definidos os mecanismos que são conjuntos de testes que podem retornar um resultado positivo ou negativo. Para cada um dos mecanismos, podem ser atribuídos modificadores, que irão determinar a ação a ser tomada caso o teste feito pelo mecanismo forneça um resultado positivo. Os mecanismos são listados de forma ordenada e são avaliados da esquerda para a direita. Caso um deles forneça um resultado positivo é tomada a ação definida pelo modificador e os outros mecanismos não são testados.

Os mecanismos definidos na RFC4408 [24] para o SPF são apresentados na Tabela III.1 e os modificadores utilizados com esses mecanismos são apresentados na Tabela III.2.

Na Figura III.6 é mostrado um exemplo de registro SPF permitindo que o servidor cujo endereço IP está associado no nome do domínio e os servidores de correio eletrônico do domínio enviem mensagens. Além disso, todas as regras utilizadas no domínio dominio.com.br também serão verificadas, permitindo que as máquinas autorizadas por ele também sejam aceitas. Todas as outras máquinas que não atendam às características anteriores são impedidas de enviar mensagens como sendo do domínio.

```
v=spf1 a mx include:dominio.com.br -all
```

Figura III.6: Um exemplo de registro SPF.

O mecanismo SPF, embora não seja utilizado diretamente para filtrar *spams*, pode ajudar a reduzí-los pois torna mais difícil o envio de mensagens com endereços falsos, caso os provedores utilizem o SPF para determinar de forma precisa as máquinas autorizadas a enviar mensagens. O uso do SPF, entretanto, não impede que um *spammer* crie vários domínios e publique registros SPF permitindo que qualquer máquina envie mensagens utilizando como remetente esses domínios. Nessa situação os servidores que utilizem o mecanismo SPF vão concluir que o servidor que está tentando enviar mensagem tem autorização, aceitando a mensagem. Os *spammers*, no entanto, não conseguirão falsificar endereços de provedores legítimos que usem o SPF, já que os mesmos podem definir políticas permitindo que somente seus servidores enviem mensagens como sendo do seu domínio. Embora um registro SPF devidamente configurado garanta que apenas os servidores de correio eletrônico do domínio especificados na política do SPF sejam aceitos como remetentes de mensagens do domínio, deve-se utilizar métodos para a autenticação dos clientes que enviam mensagens para os servidores do domínio.

III.3.3 Desafio e resposta

A idéia deste mecanismo anti-*spam* é limitar o envio de *spams*, usando como artifício o aumento do custo por mensagem enviada, fazendo com que esse custo seja maior que o

Tabela III.1: Mecanismos do SPF.

Mecanismo	Descrição
all	Retorna sempre verdadeiro. É utilizado como o último mecanismo a ser executado, para definir uma ação padrão caso nenhum dos mecanismos anteriores tenha retornado um resultado positivo.
include	Utilizado para incluir os mecanismos SPF definidos em um outro domínio especificado no parâmetro nome de domínio. Esse mecanismo é utilizado quando um servidor de um domínio também aceita que as suas mensagens sejam enviadas através de outros domínios. A sintaxe desse mecanismo é `include:<nome domínio>`.
a	Realiza uma consulta DNS ao nome do domínio para verificar se o endereço IP do servidor que está enviando a mensagem é um dos endereços IP associados ao nome do domínio indicado.
mx	Consulta, através do protocolo DNS, se o endereço IP de origem corresponde a um dos servidores de correio eletrônico do domínio, se servindo dos registros MX do DNS.
ptr	Realiza o teste do DNS reverso, apresentado na Seção III.3.1.
ip4	Define uma faixa de endereços IPv4 que estão autorizados a enviar mensagens. A sintaxe desse mecanismo é `ip4:<endereço de rede>/<máscara de sub-rede>`.
ip6	Similar ao mecanismo anterior só que utilizado para testar faixas de endereços IPv6. Sua sintaxe é a mesma do mecanismo ip4.
exists	Permite a utilização de macros para criar um determinado nome de domínio baseado em informações da mensagem. Com base no nome de domínio que foi criado, verifica se o domínio possui um registro de DNS válido. Caso o registro DNS seja válido, o mecanismo retorna um resultado positivo. A principal utilização desse mecanismo é em conjunto com as listas negras DNSBL.

Tabela III.2: Modificadores do SPF.

Modificador	Descrição
+	A mensagem é classificada como em conformidade com a política definida e o remetente é autorizado pelo domínio a enviar mensagens. Esse é o modificador padrão, tornando opcional sua definição explícita.
-	A mensagem não está de acordo com a política e o remetente não está autorizado a enviar mensagens como sendo do domínio.
?	O resultado da análise é neutro.
$\sim$	Define que o remetente provavelmente não está autorizado a enviar mensagens como sendo do domínio, mas não é feita nenhuma afirmação da sua autenticidade, permitindo assim que o servidor trate esse caso de forma diferente dos casos em que o servidor garante que o remetente está autorizado, ou não, a enviar mensagens.

lucro por mensagem. Estima-se que o lucro obtido por mensagem seja no mínimo de 0,01 centavos de dólar por mensagem [25]. Uma das primeiras propostas nesse sentido é criar um custo inicial para a criação de uma nova conta no servidor de correio eletrônico. Esse custo pode ser imposto através de uma CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*), de desafios computacionais ou do pagamento de quantias financeiras.

Um exemplo de CAPTCHA é requisitar que o usuário digite letras que são dispostas aleatoriamente em uma imagem, como a imagem mostrada na Figura III.7. Para uma pessoa, o processo de reconhecimento de caracteres é realizado de forma fácil. No entanto, para uma máquina reconhecer os caracteres da imagem, são necessários complexos algoritmos de processamento de imagens [26]. O reconhecimento pela máquina se torna ainda mais complicado quando são utilizados outros artifícios, como adicionar linhas e fundos de cores diferentes na imagem. Esse mecanismo, embora seja eficiente no sentido

de diferenciar uma máquina de uma pessoa, pode causar problemas para indivíduos com deficiência visual. Para resolver esse problema podem ser utilizados testes baseados em sons.

Figura III.7: Um exemplo de CAPTCHA.

Desafios computacionais são procedimentos ou algoritmos com um alto custo computacional. Eles são projetados de tal forma que um máquina gaste um tempo grande para sua resolução mas que possa ser resolvido precisamente, diferente das CAPTCHAs que são projetadas para não serem resolvidas por uma máquina. Como o poder de processamento das máquinas é em geral subutilizado, esses desafios computacionais podem ser resolvidos em segundo plano, fazendo com que o usuário praticamente não perceba que eles estão sendo resolvidos. A principal característica desses desafios é que eles devem ter um custo computacional elevado para sua resolução, mas um custo baixo para verificação. Dessa forma, o usuário irá gastar um tempo grande para resolvê-lo, mas o servidor pode verificar facilmente a resposta. Um desafio computacional que atende essas características é o problema de encontrar uma determinada seqüência que adicionada ao começo do cabeçalho da mensagem faça com que o seu *hash* tenha um determinado número de zeros nos bits mais significativos. Para resolver esse desafio, são necessárias tentativas de um grande número de seqüências e, para cada uma, que seja calculado o *hash* da mensagem. Já para a verificação do resultado, só é necessário adicionar a seqüência informada ao cabeçalho e calcular o seu *hash*, para verificar se os bits iniciais realmente são zero. O número de tentativas para resolver o desafio vai aumentar conforme a restrição do número de bits iniciais que devem ser iguais a zero, de forma que ajustando esse valor, pode-se definir em média em quanto tempo o desafio será realizado.

Os dois últimos tipos de custo apresentados podem ser convertidos em custos financeiros, levando-se em conta que é necessária certa quantia financeira para manter um

computador e uma pessoa e/ou uma pessoa que trabalhe resolvendo as CAPTCHAs. Para resolver as CAPTCHAs, um *spammer* terá que passar parte do seu tempo resolvendo os desafios, ou então contratar uma pessoa para resolvê-los. Nos dois casos, o tempo perdido ou o salário pago irão representar um custo financeiro, estimado da ordem de 2 centavos de dólar por cada teste que tenha que ser resolvido. Os desafios computacionais também representam um custo para o *spammer*, pois, para resolvê-los, o *spammer* utiliza a capacidade de processamento de um ou mais computadores, o que gera custos de aquisição e manutenção.

A proposta de realizar a cobrança de um determinado valor apenas inicialmente é a mais utilizada atualmente. Em muitos provedores, para um usuário criar uma nova conta ele precisa resolver uma CAPTCHA. Depois do desafio ter sido resolvido, ele não precisa resolver nenhum desafio e passa a ser limitado apenas pelo número de mensagens diárias que pode enviar. Goodman [27] faz uma análise desse método levando-se em consideração o custo inicial (C), a probabilidade de uma pessoa que recebeu um *spam* notificar o provedor de correio eletrônico que originou o mesmo (p), o número médio de dias entre o recebimento do *spam* e a reclamação (L) e, por fim, o limite diário de mensagens que podem ser enviadas (D). Nos primeiros L dias, o *spammer* conseguirá enviar D mensagens todos os dias, que é o limite máximo de mensagens, já que mesmo que a primeira pessoa que recebeu uma mensagem sua faça uma reclamação, ela só será feita, em média, após os L dias. Dessa forma, durante esse tempo, ele poderá continuar enviando *spams*. A partir do dia $L+1$, a probabilidade de reclamação para cada dia será dada pela probabilidade de qualquer uma das D mensagens enviadas serem reportadas como *spam*. Como a probabilidade de cada mensagem não ser reportada é $(1-p)$, a probabilidade de nenhuma mensagem ser reportada no dia será $(1-p)^D$. Por fim, a probabilidade q de alguma mensagem ser reportada a cada dia será justamente o complemento da probabilidade de nenhuma mensagem ser reportada, o que leva a $q = 1-(1-p)^D$. Como nos primeiros L dias o *spammer* poderá enviar D mensagens diariamente e depois disso poderá continuar enviando em média por $1/q$ dias, o número total de mensagens que serão enviadas será $LD + D/q$. Para valores pequenos de D, q será aproximadamente pD, então o número de mensagens aproximado será $LD + 1/p$. A probabilidade de reclamação é, em

geral, bem pequena, da ordem de 1/1000. Já o termo L tem ordem de grandeza de dias e D de dezenas de mensagens, fazendo com que o termo $1/p$ seja maior que o termo LD. Esse fato leva à aproximação final de $1/p$ mensagens que podem ser enviadas, não tendo o parâmetro D grande importância. Como somente foi pago um custo C inicialmente e foram enviadas aproximadamente $1/p$ mensagens, o custo por mensagem será de C/p. Supondo o custo de dois centavos de dólar do desafio, e a probabilidade p sendo 1/1000, o custo por mensagem será de 0,002 centavos, bem abaixo do lucro por mensagem. Essa é a razão pela qual esse método de cobrança inicial não conseguiu deter os *spammers*.

Uma evolução natural desse método seria impor um custo a cada mensagem enviada. Essa abordagem, no entanto, afetaria de forma negativa os usuários legítimos, que poderiam acabar desistindo de utilizar o serviço. Como uma possível solução, Goodman propõe a cobrança de custos a cada n mensagens [27], só que essa cobrança só é realizada k vezes, depois disso o usuário só é limitado pelo número de mensagens por dia que podem ser enviadas. A princípio, essa proposta não parece ser boa, mas os resultados mostram que o custo por mensagem utilizando-se o esquema de cobrança apenas nas k primeiras vezes tem um resultado bastante similar ao obtido caso a cobrança fosse feita indefinidamente, mostrando que esse método não diminui de forma considerável o custo para o *spammer*, mas é vantajoso para o usuário, que não tem que pagar para sempre os custos.

Outro sistema baseado em desafios e respostas que é utilizado atualmente funciona de forma similar às listas cinzas, apresentadas na Seção III.2.2. Diferentemente do método de cobrança de um determinado custo por mensagem ou por grupo de mensagens, nesses sistemas os usuários precisam resolver um desafio que é enviado pelo servidor do destinatário na primeira vez que enviarem uma mensagem ao destinatário. O servidor, ao receber uma mensagem para um dado par destinatário/origem que ainda não foi confirmado, guarda a mensagem e envia um desafio para o remetente da mensagem, geralmente uma CAPTCHA. Caso o remetente responda o desafio, a mensagem que tinha sido guardada é entregue ao destinatário e nas próximas vezes que o par se comunicar não será mais necessária essa verificação. Como os *spammers* utilizam mecanismos automatizados de envio de mensagens com endereços de origem forjados, acabarão não recebendo as men-

sagens contendo os desafios e, por conseqüência, suas mensagens não chegarão ao destino. Devido a essas características, esse método é bastante eficaz em relação à taxa de falsos negativos. Entretanto, usuários legítimos podem não responder aos desafios enviados. Dessa forma, mensagens legítimas acabarão sendo descartadas, gerando uma alta taxa de falsos positivos.

Capítulo IV

Sistema de Análise de *Spam* - ADES

Para realizar a análise da eficiência dos mecanismos apresentados neste trabalho foi desenvolvido o sistema ADES (**A**nálise **DE** **S**pam) que é descrito nesta seção. O sistema ADES foi desenvolvido pois não existia nenhum sistema que atendesse a todos os requisitos necessários para a análise dos mecanismos anti-*spam*. A implementação de cada um dos mecanismos anti-*spam* já existia, mas um sistema que permite a avaliação das características do mecanismos ainda não havia sido implementado. Para realizar a análise com uma grande diversidade de mensagens, foi inicialmente implementado um pote de mel que tem como objetivo divulgar endereços eletrônicos que servissem como iscas para os *spammers* enviarem uma grande quantidade de mensagens. Outro componente do sistema ADES é o módulo de verificação dos mecanismos, que é responsável por verificar cada um dos mecanismos anti-*spam* que é analisado e armazenar seus resultados. Por fim, o módulo de análise dos resultados analisa as mensagens recebidas e o resultado dos mecanismos anti-*spam* para gerar os gráficos e estatísticas. Nesta seção, são descritos os módulos implementados.

IV.1 Módulo Pote de Mel

O Pote de Mel, conforme explicado na Seção III.2.3, tem como objetivo enganar os *spammers*. Para isso são criados endereços eletrônicos que não pertencem a usuários reais e que são divulgados de forma proposital, com o objetivo de serem capturados por *spammers* e incluídos em suas listas. O Pote de Mel é comumente utilizado nas análises de mensagens não solicitadas, por permitirem uma identificação trivial de mensagens *spams*, uma vez que todas as mensagens enviadas para endereços divulgados como iscas do pote de mel são mensagens *spams* porque não são endereços eletrônicos utilizados por usuários legítimos.

Na implementação do pote de mel foram divulgados endereços eletrônicos gerados aleatoriamente e divulgados na página web principal do GTA. A cada acesso à pagina eram gerados e divulgados cinco endereços eletrônicos diferentes. A geração dos endereços eletrônicos não foi completamente aleatória, foi escolhida a geração de endereços compostos por consoantes e vogais intercaladas, com o intuito de tornar os endereços eletrônicos gerados mais similares a nomes reais e diminuir possíveis suspeitas quanto a existência dos endereços caso os *spammers* utilizem algum procedimento de verificação da validade dos endereços eletrônicos.

A divulgação dos endereços na página principal foi feita criando-se *links* na página com a mesma cor do fundo da página. Esse procedimento foi adotado para que os usuários legítimos não notassem que esses endereços estavam presentes na página. Já os mecanismos de coleta de endereços eletrônicos não levam em conta o formato ou aparência da página, realizando apenas uma busca no conteúdo da página à procura de endereços eletrônicos. Dessa forma, os mecanismos automatizados de busca acabam encontrando os endereços eletrônicos e adicionando-os às listas de envio de *spams*.

Outra característica importante do pote de mel implementado é a especificidade das informações coletadas e associadas a cada mensagem *spam*. Para cada endereço eletrônico divulgado eram armazenadas em um banco de dados as informações sobre a data de divulgação deste endereço, o endereço IP da máquina que acessou a página, o nome da página

que tinha sido acessada anteriormente, o nome da página que foi acessada e o agente utilizado para acessar a página. O nome da página acessada, neste trabalho, não foi utilizado já que os endereços foram divulgados apenas na página principal. Em futuras expansões deste trabalho, será possível distinguir em qual página cada endereço foi coletado. O nome do agente é informado pelo protocolo HTTP e tem como objetivo informar ao servidor HTTP o nome do navegador que está sendo utilizado pelo usuário. Os mecanismos de coleta de endereço tentam utilizar nomes similares ou iguais aos navegadores que são utilizados por usuários legítimos, tentando dificultar sua detecção. Baseados em todas essas informações e como cada endereço eletrônico é divulgado apenas uma vez, quando uma mensagem é enviada para algum dos endereços divulgados pelo pote de mel podem ser obtidas várias informações, como o tempo entre a divulgação do endereço eletrônico na página e o início de recebimento dos *spams*, a persistência dos endereços nas listas dos spammers e vários outros resultados que são mostrados na seção V.

IV.2 Módulo de verificação dos mecanismos

Esse módulo tem como objetivo realizar os testes de desempenho dos mecanismos anti-*spam* analisados em todas as mensagens recebidas pelo servidor de mensagens. O módulo foi implementado na linguagem Perl [28] e funciona como um servidor de políticas [29] do servidor de mensagens Postfix [30]. A arquitetura de servidores de políticas do Postfix permite a criação de módulos que recebam informações das mensagens que chegam ao servidor de correio eletrônico e definam uma ação a ser tomada com a mensagem. As principais informações sobre as mensagens que são passadas para os servidores de políticas encontram-se na Tabela IV.1. O servidor de políticas pode então realizar vários testes e/ou procedimentos e, por fim, enviar um comando ao servidor de mensagens. Os principais comandos são: descartar a mensagem, aceitar a mensagem ou adicionar informações ao cabeçalho da mensagem. A Figura IV.1 mostra o fluxo de mensagens entre o servidor de correio eletrônico e o servidor de políticas. O fluxo consiste no recebimento de uma mensagem pelo servidor de mensagens (1), na passagem de informações sobre a mensagem

ao servidor de políticas (2) e no envio de um comando pelo servidor de políticas, após realizar todos os seus procedimentos, ao servidor de mensagens. Este último fica responsável por realizar a ação informada no comando. No exemplo da figura a ação `PREPEND` é utilizada para adicionar um texto no cabeçalho da mensagem.

Tabela IV.1: Parâmetros do servidor de políticas do *Postfix*.

Parâmetro	**Exemplo**	**Descrição**
helo_name	mail.dominio.com.br	Nome utilizado para a identificação do remetente
sender	alguem@dominio.com.br	Remetente da mensagem
recipient	fulano@exemplo.com.br	Destinatário da mensagem
recipient_count	3	Número de destinatários
client_address	192.168.30.1	Endereço IP utilizado para enviar a mensagem

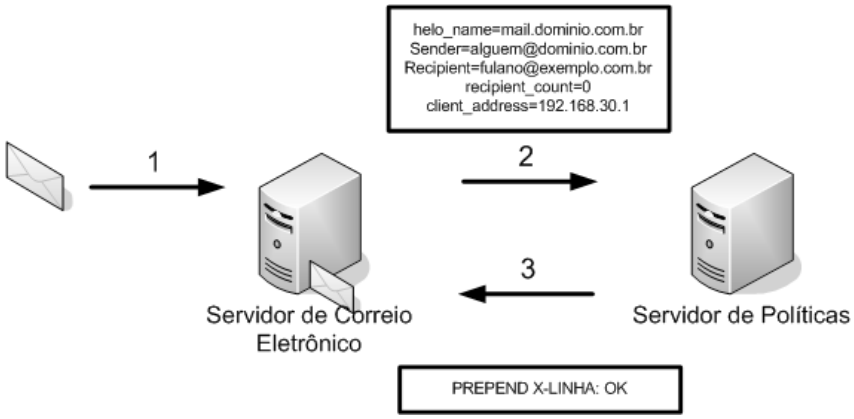


Figura IV.1: Servidor de políticas.

Na implementação realizada todas as mensagens recebidas eram analisadas através dos mecanismos de listas negras, DNS reverso e SPF. A informação do mecanismo de pesos e regras já é adicionada automaticamente ao cabeçalho da mensagem pelo servidor de correio eletrônico que está configurado para realizar este teste, não precisando o mecanismo implementado adicionar novamente essa informação. Foram escolhidos apenas esses mecanismos para a análise uma vez que são os mais utilizados. O mecanismo de filtros

bayesianos será também analisado, mas como é um mecanismo que depende do usuário para realizar o processo de treinamento, esse mecanismo não foi analisado através desse módulo.

Para o mecanismo de listas negras foi implementada a consulta às listas DNSBL e é verificado se o endereço utilizado para enviar a mensagem estava presente nas cinco[1] listas negras mais utilizadas [12]. O mecanismo de DNS reverso também é implementado e realiza a verificação se o endereço IP utilizado para enviar as mensagens possui o DNS reverso configurado corretamente. No mecanismo SPF é utilizada a biblioteca Mail::SPF [31] que já implementa a verificação do mecanismo SPF. Na verificação do mecanismo de pesos e regras é utilizado o sistema *SpamAssassin* [15].

Depois de realizar os testes de cada um dos mecanismo os resultados são armazenados adicionando-se uma linha ao cabeçalho da mensagem. Como todas as linhas do cabeçalho das mensagens, ela é composta por um nome identificador e um valor. O nome identificador utilizado foi `X-ADES-ANALYSIS` e em seguida existem vários campos separados pelo símbolo @. O primeiro campo é obrigatoriamente o endereço IP da máquina que enviou a mensagem e em seguida outros campos que definem o resultado dos mecanismos. Para cada mecanismo, o primeiro campo é código do nome do mecanismo e o segundo campo contém os resultados dos mecanismos. Essa abordagem facilita a expansão do sistema para analisar um maior número de mecanismos anti-*spam*. A seguir é ilustrado um exemplo da linha que é adiciona ao cabeçalho da mensagem:

```
X-ADES-ANALYSIS: @192.168.0.1@SPF@OK@DNSREVERSO@FALHOU
```

No exemplo acima a mensagem foi enviada através do endereço IP 192.168.0.1. Além disso, observa-se que dois mecanismos foram verificados, o primeiro com nome-código `SPF` que teve como resultado `OK` e o segundo com nome-código `DNSREVERSO` que teve como resultado `FALHOU`.

[1] As listas DNSBL consultadas foram: sbl-xbl.spamhaus.org, cbl.abuseat.org, dnsbl.sorbs.net, list.dsbl.org, bl.spamcop.net.

Como cada mecanismo pode fornecer como resultado uma ou mais informações, o campo de resultado do mecanismo não tem um padrão próprio definido, sendo sua análise e significado dependente do campo que o precede e informa o nome-código do mecanismo. Para acomodar várias informações sobre o resultado dentro de um mesmo campo são utilizados sinais de pontuação de dois pontos, para separar cada informação.

Na implementação atual, o nome dos mecanismos verificados e o nome-código do mecanismo encontram-se na Tabela IV.2. Para cada um dos mecanismos o formato dos resultados é diferente e é discutido nas seções a seguir.

Tabela IV.2: Nome-código dos mecanismos.

Nome-código	**Nome**
DNSBL	Listas Negras
SPF	*Sender Policy Framework*
RDNS	DNS reverso

IV.2.1 Listas Negras

Para os testes com o mecanismo de listas negras são consultadas as cinco listas DNSBL mais utilizadas e o resultado da consulta é armazenado. Como visto na Seção III.1.1, caso o endereço IP não esteja presente na lista negra a consulta não retorna nenhum resultado. Caso contrário a consulta retorna um endereço IP no formato 127.0.0.x, onde x é geralmente utilizado como um código para denotar o motivo pelo qual o endereço foi inserido na lista negra. O formato do resultado da verificação desse mecanismo é composto pelo nome da lista DNSBL consultada, seguido pelo tempo em milissegundos que demorou a consulta à lista e endereço IP que foi retornado na consulta a lista ou pelo símbolo #, caso o endereço não esteja presente. Como são consultadas várias listas, o resultado final do mecanismo é composto pela união dos resultados de cada uma das listas. A seguir é exemplificado o resultado de uma verificação desse mecanismo:

```
sbl-xbl.spamhaus.org:0.534:127.0.0.1:cbl.abuseat.org:0.411:#:
list.dsbl.org:0.399:#:dnsbl.sorbs.net:0.399:#:bl.spamcop.net:2206.994:#
```

Neste exemplo observa-se que o endereço IP consultado estava presente somente na lista `sbl-xbl.spamhaus.org` e que a consulta mais demorada foi à lista `bl.spamcop.net`, que consumiu mais de dois segundos.

IV.2.2 *Sender Policy Framework* (SPF)

Para realizar a verificação desse mecanismo são passados à biblioteca *Mail::SPF* o endereço IP de origem da mensagem e o endereço eletrônico do remetente da mensagem. A biblioteca realiza o procedimento de verificação do SPF conforme descrito na Seção III.3.2 e retorna um resultado de acordo com o modificador do mecanismo SPF que teve um resultado positivo, conforme mostrado na Tabeae III.2. O resultado retornado pela biblioteca de acordo com cada um dos modificadores é mostrado na Tabela IV.3. Além desses resultados, existe ainda o resultado `none` que é retornado caso o domínio não tenha definido uma política SPF e `error` caso ocorra algum erro durante o processo. Um exemplo do resultado desse mecanismo é mostrado a seguir:

```
neutral:3031.575
```

Nesse exemplo o resultado da verificação do SPF foi neutro e a consulta demorou cerca de três segundos.

O resultado final da verificação desse mecanismo é composto pelo resultado fornecido pela biblioteca seguido pelo tempo em milissegundos gasto para realizar a consulta.

IV.2.3 DNS reverso

Para verificação do DNS reverso, é feita inicialmente uma consulta DNS reversa com o endereço IP utilizado para enviar a mensagem. Caso essa consulta não retorne um nome

Tabela IV.3: Resultados da biblioteca *Mail::SPF*.

Modificador	Resultado
+	*pass*
-	*fail*
~	*softfail*
?	*neutral*

válido, produz-se o código `no-rdns` no resultado. Caso exista o nome, então verifica-se o IP associado a esse nome. Caso não exista o IP associado ao nome, utiliza-se o código `rdns-no-ip` no resultado. Caso o IP exista, mas não seja igual ao IP utilizado para enviar a mensagem, usa-se o código `rdns-ip-mismatch`. Por fim, caso os dois endereços IP sejam iguais, usa-se o código `ok`. Na verificação convencional do DNS reverso só é verificado se o endereço IP possui um nome associado, mas optou-se por realizar os outros testes para permitir a obtenção de mais dados e permitir usar essas informações adicionais em análises futuras.

Como resultado final desse mecanismo, são agrupados o nome associado ao endereço IP utilizado para enviar as mensagens, o IP associado a esse nome, o código do resultado conforme descrito anteriormente e o tempo em milissegundos que a consulta demorou. Um exemplo de resultado desse mecanismo é mostrado a seguir:

```
IGLD-80-230-79-56.inter.net.il:80.230.79.56:ok:3.005
```

Nesse exemplo, o nome associado ao endereço IP que foi utilizado para enviar a mensagem é *IGLD-80-230-79-56.inter.net.il* e o endereço IP associado a esse nome é 80.230.79.56. Como nesse caso o código do resultado foi `ok`, o endereço IP utilizado para enviar a mensagem foi igual ao endereço IP 80.230.79.56. Por fim, pode-se notar que a consulta demorou cerca de três milissegundos para ser realizada.

IV.2.4 Pesos e Regras

Para realizar o teste com esse mecanismo foi escolhido o sistema de pesos e regras mais utilizado atualmente, o *SpamAssassin* [15]. As informações desse mecanismo são adicionadas automaticamente pelo servidor de correio eletrônico, que realiza a verificação de todas as mensagens recebidas através do *SpamAssassin*. Ao final da análise de cada mensagem são adicionadas três linhas ao cabeçalho da mensagem mostrando quais as regras que tiveram um resultado positivo, o somatório total dos pesos e a decisão final do mecanismo se a mensagem foi classificada, ou não, como *spam*. As linhas adicionadas ao cabeçalho das mensagens têm um formato similar ao exemplo abaixo:

```
X-Spam-Status: Yes, hits=60.0 tagged_above=-999.0 required=3.0
tests=BAYES_99,
BR_ADJUST_2, BR_CALL_KITNET, BR_CLIQUE_AQUI, BR_FROM_KEYS,
BR_KITNET_URI, BR_SPAMMER_URI, DNS_FROM_RFC_POST, DOMAIN_RATIO,
HTML_90_100, HTML_IMAGE_ONLY_12, HTML_IMAGE_RATIO_02, HTML_MESSAGE,
MIME_BOUND_DD_DIGITS, MIME_HTML_ONLY, MIME_HTML_ONLY_MULTI,
MPART_ALT_DIFF, MSGID_SPAM_CAPS, MSGID_YAHOO_CAPS,
RAZOR2_CF_RANGE_51_100, RAZOR2_CHECK, RCVD_BY_IP,
RCVD_DOUBLE_IP_SPAM, RCVD_HELO_IP_MISMATCH, RCVD_IN_BL_SPAMCOP_NET,
RCVD_IN_DSBL, RCVD_IN_NJABL_PROXY, RCVD_IN_SORBS_WEB, RCVD_IN_XBL,
RCVD_NUMERIC_HELO, SPF_FAIL, URIBL_OB_SURBL, URIBL_WS_SURBL
X-Spam-Level: ************************************************************
X-Spam-Flag: YES
```

A primeira linha mostra inicialmente se a mensagem foi classificada, ou não, como *spam*, seguido pelo somatório dos pesos no parâmetro *hits*. O parâmetro *tagged_above* determina a partir de qual valor do somatório dos pesos a mensagem vai ser marcada como tendo sido analisada ou não. Em seguida, o parâmetro *required* determina o limite do somatório dos pesos a partir do qual as mensagens são classificadas como *spam*. No

parâmetro *tests* são descritos os códigos de todas as regras com resultado positivo. Uma descrição completa de todos os testes pode ser encontrada em [16].

A segunda linha, que é iniciada por `X-Spam-Level`, mostra de forma simbólica o somatório dos pesos, sendo que cada asterisco representa uma unidade do somatório dos pesos. No caso do exemplo, como o somatório dos pesos foi 60, existem 60 asteriscos nessa linha.

A terceira linha contém uma informação que já aparece na primeira linha, mostrando de forma resumida o resultado final da classificação da mensagem como *spam* ou legítima. Caso o mecanismo tenha decidido que a mensagem é um *spam*, o resultado que aparece nessa linha é `YES`. Caso contrário, o resultado é `NO`. As informações sobre o tempo de análise de cada uma das mensagens analisadas não é incluído no cabeçalho da mensagem, sendo apenas acessível através do arquivo de *log* do *SpamAssassin*.

IV.3 Módulo de análise dos resultados

Este módulo tem como objetivo realizar a análise dos resultados dos mecanismos que foram testados pelo módulo de verificação dos mecanismos, além de gerar estatísticas sobre a quantidade, o tamanho, a distribuição temporal e outros resultados que serão mostrados no capítulo V.

Para realizar a análise dos resultados dos mecanismos anti-*spam* é necessário acessar a caixa postal dos usuários e verificar no cabeçalho de cada mensagem recebida as informações que foram adicionadas pelo módulo de verificação dos mecanismos. Assim, a implementação deste módulo foi feita utilizando a linguagem Perl, já que essa linguagem dispõe de várias rotinas e procedimentos de manipulação de dados e ferramentas poderosas de análise de texto, como expressões regulares, que permitem encontrar um determinado padrão em um texto ou mensagem. Para analisar as mensagens recebidas pelos usuários foi utilizada a biblioteca *Mail::Box* [32] disponível gratuitamente e que possibilita o acesso às mensagens da caixa postal dos usuários de forma fácil.

Para cada uma das mensagens recebidas, primeiramente é feita uma busca no cabeçalho da mensagem procurando a linha que inicia com `X-ADES-ANALYSIS`, que foi adicionada à mensagem pelo módulo de Verificação dos mecanismos. Depois, cada um dos campos é separado e o resultado de cada um dos mecanismos armazenado é analisado. O mesmo procedimento é realizado com a linha `X-Spam-Flag`, que é a linha adicionada pelo mecanismo anti-*spam* de pesos e regras.

Para calcular os índices de falsos positivos e falsos negativos de cada um dos mecanismos é necessário saber *a priori* se a mensagem é legítima ou *spam*, para comparar com o resultado do mecanismo. Para isso todas as mensagens recebidas pelo autor foram classificadas anteriormente de forma manual. Com base nessa informação, pode-se então calcular a taxa de falsos positivos e falsos negativos de cada um dos mecanismos.

Esse módulo também gera outras estatísticas sobre as mensagens, como a distribuição temporal, distribuição por faixas de endereço IP e por países. Para determinar a distribuição por países foram utilizadas as bibliotecas gratuitas *IP::Country* [33] e *Geography::Countries* [34] que possuem uma base de dados com a informação sobre a localização geográfica das faixas de endereço IP e informações sobre os países.

Também são realizadas análises em relação ao uso do pote de mel, fazendo a comparação entre os endereços divulgados como iscas no pote de mel e os endereços que receberam mensagens *spam*. Para realizar essas estatísticas, busca-se inicialmente todas as mensagens que foram enviadas para endereços divulgados no pote de mel. Depois das mensagens serem identificadas, verifica-se quantas mensagens cada um dos endereços divulgados recebeu, o tempo entre a coleta dos endereços e o envio das mensagens e também se a coleta e o envio foram feitos a partir do mesmo endereço IP.

Capítulo V

Testes e análise dos resultados

Neste capítulo é discutida a metodologia utilizada para a realização dos testes e em seguida é apresentada a análise dos resultados dos testes que foram realizados com o sistema ADES.

V.1 Metodologia

Para a análise da eficácia dos mecanismos foi feita uma classificação manual prévia de todas as mensagens recebidas que correspondem a *spams* e mensagens legítimas. As mensagens legítimas consideradas na análise foram todas as mensagens legítimas recebidas pelo autor deste trabalho. Já para as mensagens *spams* são consideradas as mensagens recebidas pelo autor e também as mensagens *spams* que possuíam um destinatário que não representava um usuário válido no domínio. Dessa forma, o autor recebia tanto os *spams* que eram direcionados para ele quanto os *spams* que eram enviados para endereços eletrônicos divulgados pelo pote de mel e também *spams* enviados através de ataques de dicionário ou de força bruta. Essa metodologia permitiu que fosse obtida uma grande quantidade de mensagens *spams* para realizar a análise. Os testes foram realizados dauante o período de 04/10/2006 a 24/11/2006 e durante este período foram recebidas 332 mensagens legítimas e 15.178 mensagens *spams*.

Para os filtros bayesianos o teste utiliza o mecanismo de filtros bayesianos do programa de correio eletrônico Mozilla Thunderbird [35]. Inicialmente metade das mensagens legítimas e metade dos *spams* foram separados para serem usados no treinamento do filtro. Na outra metade aplicou-se o filtro e analisou-se os falsos positivos e negativos. Os dois grupos de mensagens foram separados através da ordem cronológica de chegada, sendo as mais antigas utilizadas para treinar o filtro, simulando o cenário em que o filtro é inicialmente treinado com mensagens recebidas e utiliza-se o filtro para classificar mensagens recebidas.

V.2 Resultados

Os resultados são apresentados em três seções diferentes. Inicialmente serão apresentados os resultados de falsos positivos, falsos negativos e tempo para verificação dos mecanismos anti-*spam* avaliados. Em seguida serão mostradas características fundamentais em que os *spams* são diferentes das mensagens legítimas e, por fim, serão mostrados os resultados do sistema de pote de mel implementado.

V.2.1 Resultados dos mecanismos

O módulo de análise dos resultados calcula a taxa de falsos positivos e falsos negativos de cada um dos mecanismos analisados. Na Tabela V.1 encontra-se o resultado em número de mensagens para cada um dos mecanismos. Já a Figura V.1 mostra os percentuais de falsos positivos e falsos negativos dos mecanismos.

Para o mecanismo *SPF* foram considerados os falsos positivos e negativos apenas levando em consideração as mensagens cujo remetente pertencia a um domínio que publicava a informação de política do mecanismo SPF. Das 332 mensagens legítimas, em apenas 24 (7,2%) mensagens foi possível analisar o mecanismo SPF. Já para os *spams*, em 2.262 (14,9%) mensagens foi possível realizar a análise. A adoção maior do *SPF* no caso das mensagens *spams* é um artifício para que os *spammers* passem no teste do mecanismo

Tabela V.1: Resultado da análise dos mecanismos.

Mecanismo	Falsos Positivos	Falsos Negativos	Universo de Mensagens	
			Spams	Legítimas
Listas Negras	12	2.236	15.178	332
SPF	1	668	2.262	24
DNS Reverso	80	9.253	15.178	332
Pesos e Regras	1	2.692	15.178	332
Filtros Bayesianos	4	2.299	7.704	173

SPF, diminuindo a probabilidade do *spam* ser classificado como tal.

A taxa de falsos positivos para os mecanismos de listas negras e DNS reverso foi consideravelmente alta, chegando a quase 1/4 das mensagens para o mecanismo de DNS reverso. Dessa forma, a utilização desses mecanismos como forma individual de classificar uma mensagem como *spam* ou não pode gerar uma taxa de falsos positivos muito grande, causando uma grande insatisfação dos usuários. O mecanismo de pesos e regras obteve uma taxa de falsos positivos reduzida, sendo apenas uma mensagem classificada incorretamente. Para o mecanismo SPF a compração da taxa de falsos positivos com a taxa de outros mecanismos é prejudicada devio à pequena quantidade de amostras.

Em todos os mecanismos a taxa de falsos negativos foi superior a 14%, chegando a quase 61% para o mecanismo de DNS reverso, o que mostra sua grande ineficiência. A taxa alta de falsos negativos para o DNS reverso é provavelmente causada uma vez que os *spammers* utilizam máquinas de terceiros para enviarem *spams* e essas máquinas possuem o DNS reverso configurado corretamente. Já os falsos positivos do mecanismo de DNS reverso acontecem pois muitos administradores não configuram corretamente o DNS reverso de seus domínios. O mecanismo de verificação do DNS reverso foi criado já que os *spammers* não configuravam o DNS reverso corretamente para dificultar o rastreamento de suas máquinas, mas como agora a maioria dos *spams* é enviado por máquinas de terceiros

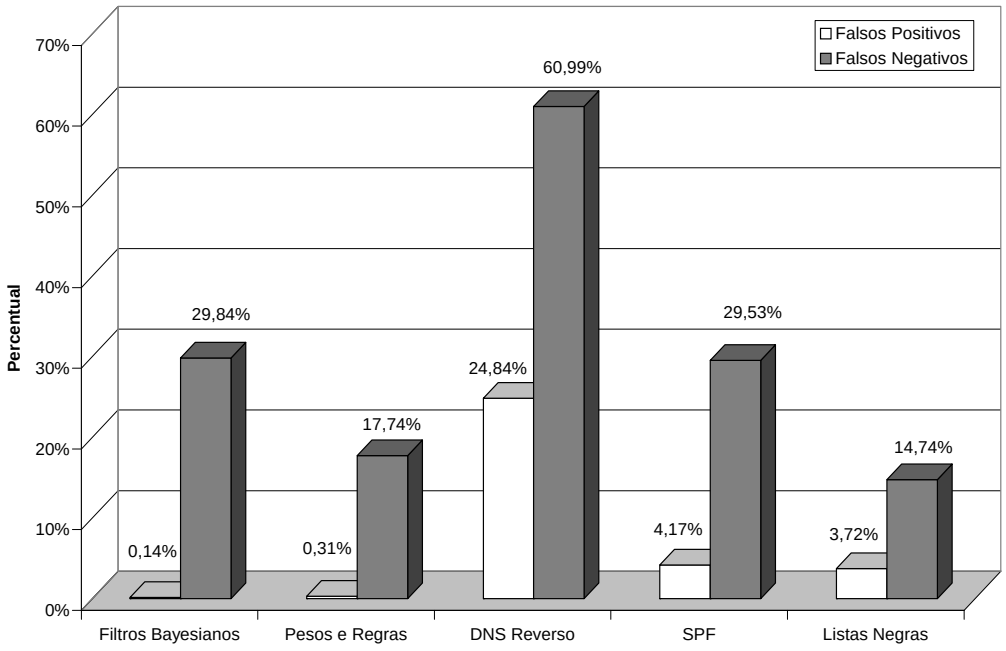


Figura V.1: Percentual de falsos positivos e falsos negativos.

que possuem o DNS reverso configurado corretamente, a taxa de falsos negativos desse mecanismo é muito grande.

No teste do filtro bayesiano foram utilizadas 173 mensagens legítimas e 7.704 *spams* para o treinamento do filtro. O mesmo número de mensagens foi utilizado para aplicar o filtro que tinha sido treinado anteriormente. O filtro classificou 4 das mensagens legítimas como *spam*, o que equivale a uma taxa de falsos positivos de 0,14%. Já para as mensagens *spams* a taxa de falsos negativos foi de 29,84%, o que equivale a 2.299 mensagens.

A Figura V.2 apresenta o gráfico da função de distribuição cumulativa (CDF) do tempo que a verificação de cada mecanismo demorou para ser executada. Esse tempo é afetado por fatores como a condição da rede e a configuração do computador, mas como todas as verificações são realizadas utilizando o mesmo computador e a mesma conexão de rede à Internet, os resultados podem ser comparados. Os mecanismos que obtiveram melhor resultado foram os mecanismos de listas negras e DNS reverso, levando menos de um segundo em 90% dos casos. Isso ocorre já que em ambos são necessárias apenas consultas DNS. Embora os mecanismos de listas negras e DNS reverso utilizem apenas consultas DNS, o último tem um resultado melhor pois o servidor de correio eletrônico já

realiza a consulta do DNS reverso do endereço IP que está enviando a mensagem. Dessa forma, quando o mecanismo do DNS reverso realiza novamente a consulta, o resultado já encontra-se no *cache*, reduzindo o tempo de consulta. Dentre as listas negras consultadas, a lista negra 1 que representa a lista `sbl-xbl.spamhaus.org` teve o melhor resultado. Esse resultado se deve ao fato dessa lista ser a mais utilizada, aumentando as chances das consultas à lista serem respondidas por servidores DNS intermediários que já possuem a informação guardada em seu *cache*. O mecanismo SPF obteve um resultado ligeiramente menor, provavelmente devido à maior complexidade da análise das políticas SPF. Já o mecanismo de pesos e regras obteve o pior resultado de todos, levando até 12 segundos em 90% dos casos. O menor desempenho desse mecanismo pode ser explicado já que esse mecanismo é baseado em análise de conteúdo, o que requer um processamento maior. O mecanismo de filtros bayesianos não foi analisado em relação ao tempo de verificação, pois essa informação não é disponibilizada pelo programa que implementa o mecanismo.

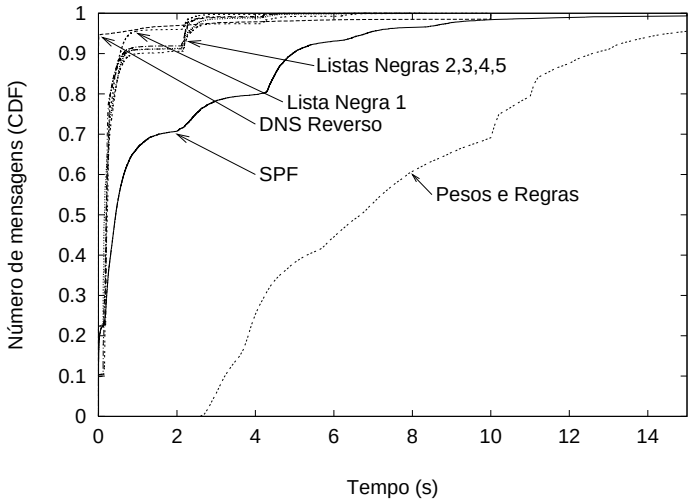


Figura V.2: Distribuição do tempo de verificação de cada mecanismo.

V.2.2 Características dos *spams*

O *spam* tem características diferentes das mensagens legítimas, como discutido nas Seções III.2.1 e III.2.4. Os *spammers* comportam-se e possuem uma relação diferente dos usuários legítimos. Os *spammers* tentam enviar a mensagem *spam* para o maior número

possível de destinatários. Por sua vez, os usuários legítimos trocam mais mensagens durante o período de trabalho e durante discussões por correio eletrônico.

A Figura V.3 mostra a função de distribuição cumulativa da quantidade de mensagens legítimas e *spams* recebidas. Para os *spams* pode-se notar que a taxa de crescimento não sofre variações com o tempo. O comportamento das mensagens legítimas é diferente, pois ocorrem períodos de maior troca de mensagens e períodos com menos trocas de mensagens. Na semana do dia 21 a 28 de outubro, podemos ver um grande aumento na quantidade de mensagens recebidas, que foi causado pela discussão de um projeto.

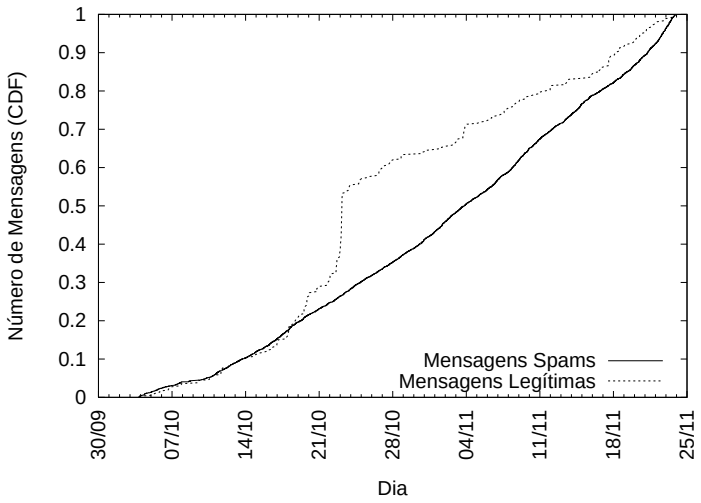


Figura V.3: Distribuição temporal das mensagens.

A Figura V.4 mostra a função de distribuição cumulativa do número de mensagens recebidas em função das horas do dia. Pode-se notar que durante o período da madrugada são trocadas poucas mensagens legítimas. No período de uma a duas horas da manhã são trocadas aproximadamente 10% de todas as mensagens. Já no período de duas até as sete horas da manhã a quantidade de mensagens trocadas é muita baixa, resultando em um baixo crescimento do número acumulado de mensagens. Em contrapartida, para os *spams*, o envio de mensagens é constante, independente da hora do dia, uma vez que as mensagens são enviadas por mecanismos automatizados, que podem ser executados continuamente.

A Figura V.5 mostra o gráfico da função de distribuição cumulativa do tamanho das

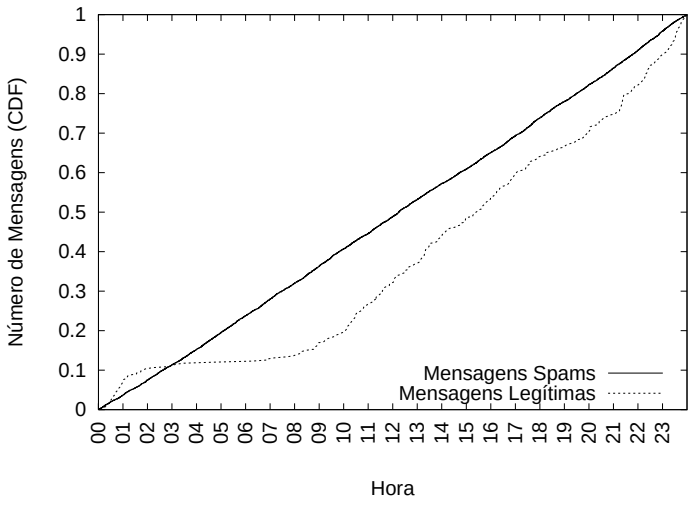


Figura V.4: Distribuição por hora das mensagens.

mensagens. Conforme demonstrado no gráfico, as mensagens *spams* geralmente possuem um tamanho menor que as mensagens legítimas, embora a diferença para mensagens com menos do que 20 kilobytes não seja muito significativa. Uma das preocupações iniciais dos *spammers* era a criação de mensagens com um tamanho pequeno, para enviá-las em menos tempo. Assim, é possível enviar a mensagem para um número maior de destinatários em um menor intervalo de tempo. Atualmente, muitos *spammers* estão utilizando imagens para tentar enganar os mecanismos que realizam análise por conteúdo, o que acaba tornando as mensagens maiores do que as mensagens *spam* apenas com texto. Como muitas mensagens legítimas são apenas de texto, porém longas, o tamanho da mensagem fica comparável ao tamanho do *spam* que possui uma imagem. Dessa forma, a distribuição das mensagens *spams* e legítimas para tamanhos pequenos é bastante similar. Este resultado se diferencia de outros resultado já apresentados na literatura, como em [19]. Isso ocorre pois os testes em [19] foram realizados em 2004 e as mensagens *spams* com imagens ainda não eram comuns.

Outra característica que difere as mensagens legítimas dos *spams* é a distribuição dos remetentes através de faixas do endereço IP. A Figura V.6 mostra a função de distribuição cumulativa do número de mensagens em função da faixa de endereços IP. Como pode ser observado no gráfico, as mensagens *spams* tem uma distribuição mais dispersa, enquanto

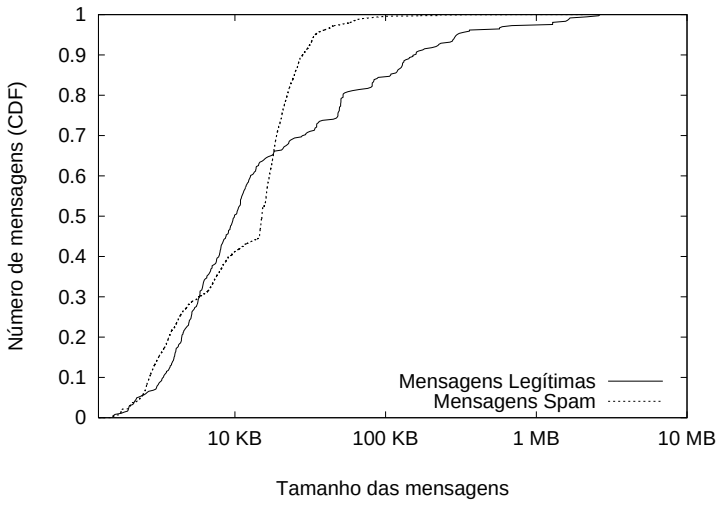


Figura V.5: Distribuição por tamanho das mensagens.

que para as mensagens legítimas existem concentrações em determinadas faixas. Na faixa *146.164.0.0/16* existe uma grande concentração de mensagens legítimas, justamente por essa faixa pertencer à Universidade Federal do Rio de Janeiro e serem trocadas muitas mensagens entre pessoas da universidade.

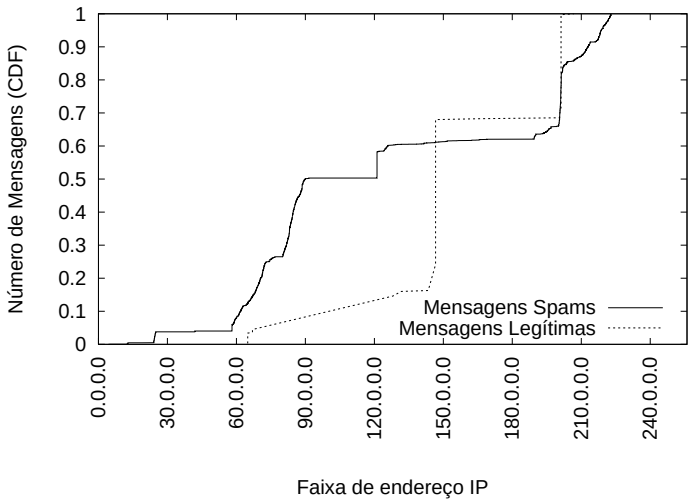


Figura V.6: Distribuição por faixas de endereços IP das mensagens.

A partir do endereço IP pode-se também determinar qual o país em que o endereço IP está localizado. Na Tabela V.2 são mostrados os 10 países que mais enviaram *spams*. A Figura V.7 mostra a distribuição geográfica dos endereços IP que enviam *spams*. Como

pode ser visto na figura, existe uma maior concentração dos endereços IP que enviam *spam* na América do Norte e Europa.

Tabela V.2: Países responsáveis pelo envio de *spams*.

País	**Número de *spams***
Estados Unidos	3005
Brasil	2344
China	1294
Israel	560
República da Coréia	551
Alemanha	550
França	471
Polônia	437
Espanha	389
Reino Unido	370

V.2.3 Análise do pote de mel

Para o pote de mel implementado analisou-se o tempo entre a divulgação do endereço eletrônico na página e o recebimento do primeiro *spam*. A Figura V.8 mostra a função de distribuição cumulativa do número de mensagens recebidas em função do tempo entre a divulgação do endereço e o recebimento primeiro *spam*. Observa-se que em quase 90% dos casos os *spams* só começaram a ser enviados após uma semana. Esse resultado mostra que mesmo que o endereço eletrônico deixe de ser divulgado, ele ainda continuará a receber *spams* durante um tempo grande, já que o tempo entre o processo de coleta e o envio dos *spams* é em geral bem longo.

Outra característica importante a ser observada é o número de mensagens recebidas por cada um dos endereços divulgados no pote de mel. Como cada endereço é divulgado apenas uma vez, sendo trocados a cada visita à página, esse teste tem como objetivo medir

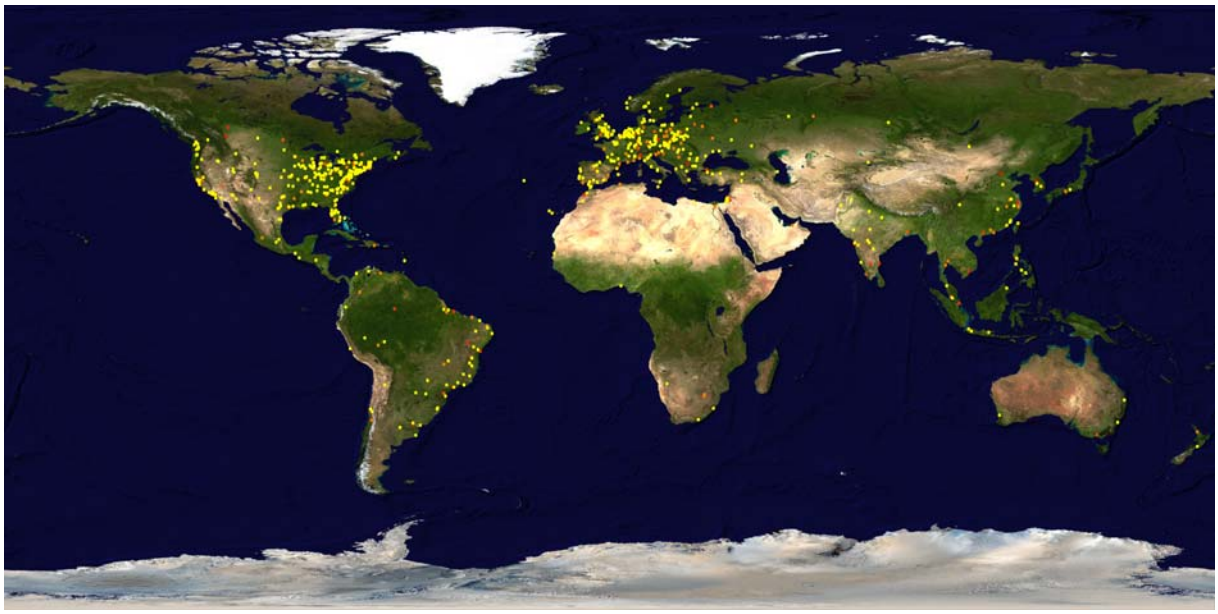


Figura V.7: Distribuição geográfica dos endereços IP que enviam *spam*.

a persistência dos endereços eletrônicos nas listas dos *spammers*. A Figura V.9 apresenta a função de distribuição cumulativa do percentual dos endereços em função do número de mensagens recebidas. Esse gráfico mostra que 86% dos endereços divulgados receberam apenas um *spam* e 97% no máximo dois *spams*. Dessa forma, pode-se inferir que as listas dos *spammers* estão constantemente sendo atualizadas. Comparando com o resultado anterior é de se esperar que após a remoção de endereços eletrônicos de páginas demore um tempo inicial da ordem de algumas semanas para começar a redução do número de *spams* recebidos e que depois de algumas semanas a quantidade de *spams* reduza bastante, já que o tempo de permanência dos endereços eletrônicos nas listas é baixo.

Outra característica do processo de coleta de endereços e envio das mensagens que pode ser analisada é se estes dois processos são realizados na mesma máquina. A Figura V.10 mostra o gráfico da função de distribuição cumulativa do percentual em função da distância[1] entre o endereço IP que realizou o processo de coleta e o endereço IP que realizou o processo de envio do *spam*. Observa-se do gráfico que em cerca de 80% dos casos a dis-

[1] A distância entre dois endereços IP foi definida como o módulo da diferença entre a representação em decimal dos endereços IP. Para converter o endereço IP que é geralmente escrito no formato $x.y.z.w$, simplesmente realizamos a operação: $x * 256 * 256 * 256 + y * 256 * 256 + z * 256 + w$.

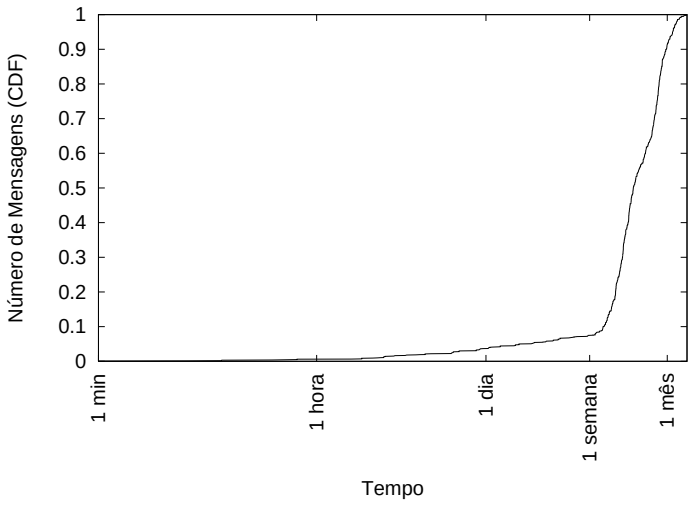


Figura V.8: Distribuição do tempo entre coleta e envio.

tância é maior do que 10^9. Se for computada a diferença entre os dois endereços da forma $x.y.0.0$ e $x.(y+1).255.255$, que representam os extremos de duas redes /16 adjacentes, representando o melhor caso entre a distância de duas redes grandes, a distância entre os dois endereços é 131.071. Esse fato sugere que as máquinas utilizadas na coleta e envio geralmente não se encontram nem na mesma rede. Essa característica pode ser causada pela execução paralela dos mecanismos de coleta e envio em máquinas separadas ou então a utilização de máquinas zumbis que podem parar de funcionar e o processo de envio de *spams* continua por outra máquina.

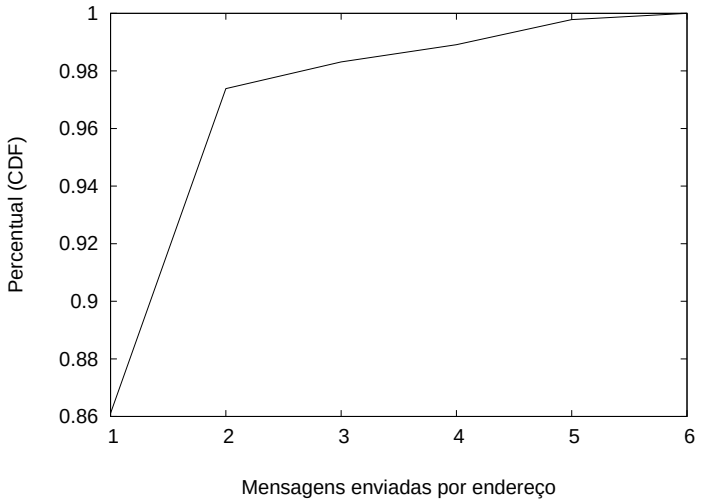


Figura V.9: Distribuição dos endereços em função do número de mensagens recebidas.

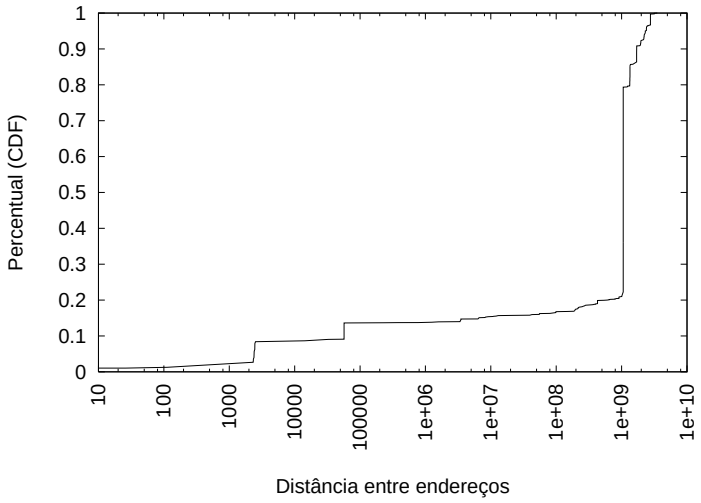


Figura V.10: Distribuição da distância entre os endereços IP.

Capítulo VI

Conclusão

Não existe hoje nenhum indício que permita inferir que a atividade de enviar *spams* diminuirá nos próximos anos. Ao contrário, os *spammers* vêm se especializando e usando técnicas cada vez mais elaboradas para burlar os sistemas anti-*spam*. Vale ressaltar que os sistemas anti-*spam* estão em constante evolução, pois, a cada novo mecanismo anti-*spam* criado, os *spammers* desenvolvem novas técnicas para enganá-los e permitir a passagem das mensagens não solicitadas.

Este trabalho teve como objetivo mostrar o funcionamento dos mecanismos anti-*spam* existentes e realizar testes para avaliar o desempenho de cada um dos mecanismos mais utilizados. Os mecanismos anti-*spam* dividem-se em três grandes grupos: os de filtragem simples, os com auto-aprendizado e os baseados em verificação da origem. Os mecanismos de filtragem simples têm a vantagem da simplicidade. No entanto, requerem atualizações periódicas das regras e dados que utilizam para classificar as mensagens. Para resolver este problema existem os mecanismos com auto-aprendizado, com capacidade de aprendizado e adaptação aos *spams*. Já os mecanismos de verificação da origem verificam a autenticidade da origem, verificando tanto se o endereço de origem não é forjado quanto se a origem é um mecanismo automatizado para enviar mensagens ou um usuário legítimo.

Para realizar a análise da eficiência dos mecanismos foi desenvolvido o sistema ADES (**A**nálise **DE** **S**pam) pois não existia nenhum sistema que atendesse a todos os requisitos

necessários para a análise dos mecanismos anti-*spam*. O sistema ADES permite a análise dos falsos positivos, falsos negativos e tempo de análise dos diferentes mecanismos anti-*spam*. Além disso, o sistema possui uma arquitetura modular e flexível, permitindo a análise de diferentes características das mensagens e do processo de envio de *spams*. O sistema ADES também possui um pote de mel para enganar os *spammers*, divulgando endereços eletrônicos que não pertencem a usuários legítimos e servem apenas para receber *spams*.

Os testes dos mecanismos mostraram que a taxa de falsos negativos de todos os mecanismos é superior a 14%. Essa alta taxa de falsos negativos implica em uma grande perda de tempo dos usuários como os *spams*. Outro aspecto importante observado foi a alta taxa de falsos positivos, fazendo com que mensagens legítimas acabem sendo classificadas como *spam*.

O sistema desenvolvido também permite analisar características dos *spams*, ajudando a entender as técnicas utilizadas pelos *spammers* para enviar as mensagens. A análise da distribuição temporal das mensagens mostrou que os *spammers* enviam as mensagens de forma contínua durante todo o dia. Os usuários legítimos, no entanto, enviam poucas mensagens durante o período da madrugada. O tamanho das mensagens também possui uma distribuição diferente, sendo os *spams* menores.

Os procedimentos de coleta de endereços de correio eletrônico e envio de mensagens também podem ser analisados através do sistema ADES. Os resultados mostram importantes características deste processo, como o tempo entre a coleta e o envio da mensagens que é da ordem de algumas semanas e o tempo baixo de permanência dos endereços nas listas dos *spammers*. O envio e a coleta são feitos, na maioria dos casos, de endereços IP diferentes, indicando que os procedimentos de coleta e envio são realizados por máquinas diferentes e o *spammer* tem o controle de uma grande quantidade de máquinas.

Na luta contra os *spams*, deve-se levar em consideração que a maioria dos usuários da Internet não tem formação técnica em computação, possuindo uma capacidade limitada para gerenciar e configurar seus computadores. Os resultados dos testes também

mostraram que os mecanismos anti-*spam* atuais possuem uma baixa eficiência. Portanto, é fundamental que se construam sistemas o mais independentemente possível da intervenção humana. Uma das propostas nesse sentido é a utilização de sistemas autônomos [36] para combater os *spams*. A idéia é fazer com que os sistemas anti-*spam* sejam capazes de se adaptar, sem a intervenção humana, às novas técnicas que vão sendo criadas pelos *spammers*. Tais sistemas devem possuir, além da característica de auto-aprendizado já encontrada em alguns sistemas atuais, as propriedades de auto-gerenciamento, auto-manutenção, auto-configuração e auto-recuperação. Neste novo paradigma, já existem propostas que utilizam mecanismos bio-inspirados [37] criando um sistema imunológico artificial capaz de combater os *spams*. O sistema imunológico humano se baseia na identificação e na destruição de agentes que causem mal ao sistema, chamados de patógenos, que podem ser vírus, bactérias ou fungos. O sistema imunológico para detectar e neutralizar os patógenos utiliza linfócitos, que são células capazes de identificar os patógenos e destruí-los. A idéia do mecanismo bio-inspirado proposto por Oda e White é utilizar linfócitos virtuais, que de forma análoga aos linfócitos do sistema imunológico humano, possuem receptores capazes de se ligar a antígenos, que representam as características do *spam* [37]. Um antígeno é uma partícula capaz de iniciar a produção de um anticorpo específico. A proposta de adoção desse modelo se baseia no fato de que um *spam* é similar a um resfriado. Ele não causa efeitos graves, mas é um mal que incomoda as pessoas e está em constante evolução.

Uma nova forma de mensagem não solicitada que está surgindo é o *spam* através de serviços de voz sobre IP (VoIP) [38]. Os *spams* de VoIP, também chamados de SPITs (*Spam over Internet Telephony*), consistem de mensagens, em sua maioria de conteúdo publicitário, enviadas em difusão através de serviços de telefonia IP. Espera-se que, em um curto espaço de tempo, o volume de *spams* dessa natureza cresça em virtude do aumento do número de usuários dos sistemas de telefonia sobre IP. Assim como as mensagens não solicitadas de texto, os *spams* de VoIP têm como atrativo para o *spammer* um custo bem menor do que o custo do envio de mensagens através da rede telefônica convencional. O telemarketing e as gravações publicitárias são formas de *spam* usando a rede telefônica convencional. Na rede convencional de telefonia, só pode existir uma

única comunicação telefônica ativa por linha telefônica e, portanto, para se enviar várias mensagens de telemarketing e gravações publicitárias ao mesmo tempo são necessárias diversas linhas telefônicas. Por outro lado, na telefonia VoIP, pela própria característica da Internet, um *spammer* pode realizar simultaneamente um grande número de ligações utilizando apenas um acesso à Internet. Para combater os *spams* de VoIP, a maioria dos mecanismos anti-*spam* deve ser modificada, já que muitos desses mecanismos se baseiam no conteúdo do texto da mensagem para filtrá-las. A análise por conteúdo se torna muito difícil nos *spams* de VoIP, pois exige a execução de algoritmos de reconhecimento de voz. Atualmente, esses algoritmos demandam um alto custo computacional e não são muito eficientes. Além disso, ao contrário de uma mensagem de texto que é recebida por um servidor e depois encaminhada ao destinatário para que ele a leia quando quiser, uma chamada telefônica ocorre em tempo real, ou seja, é necessário que o usuário atenda a chamada para que ela se inicie. Sendo assim, a análise do conteúdo de um *spam* de VoIP tem que ser feita no momento da comunicação, o que torna o mecanismo ineficiente, uma vez que o usuário já atendeu à ligação e escutou um determinado trecho da propaganda. Por esse fato, os *spams* de VoIP tendem a se tornar ainda mais incômodos do que os *spams* de texto para os usuários, que precisam parar suas atividades para atender a chamada e só depois descobrir que era uma mensagem não solicitada.

As mensagens não solicitadas em forma de vídeo também estão surgindo e assim como os *spams* de VoIP são difíceis de ser classificadas, já que para analisar o conteúdo das mensagens de vídeo são necessárias técnicas de processamento e reconhecimento de padrões de imagem. Os *spams* de vídeo estão começando a aparecer em sítios da Internet que, sem a permissão do usuário, carregam e mostram um determinado vídeo, que na maioria das vezes possui um conteúdo publicitário. Dessa forma, durante a exibição do vídeo, a atenção do usuário é desviada para o anúncio, devido aos sons e aos movimentos. A identificação dos *spams* de vídeo é complexa, pois em muitas ocasiões um sítio contém um vídeo que é do interesse do usuário e, portanto, não deve ser considerado como um *spam*.

O surgimento dos *spams* de voz e vídeo irão tornar muito difícil a análise por conteúdo.

Dessa forma, mecanismos que não se baseiam na análise de conteúdo, como o mecanismo de padrões sociais e comportamento de tráfego são fortes candidatos para a solução desse problema que está começando a surgir.

Está claro que o envio de *spams* se tornou uma atividade financeira atrativa para os *spammers* tanto para anunciar produtos e serviços quanto pela possibilidade de enriquecimento ilícito através de fraudes. Para tanto, os *spammers* estão se especializando cada vez mais. Neste sentido, prevê-se uma batalha interminável entre os *spammers* e os desenvolvedores de sistemas anti-*spam* e a criação de medidas legais para punir os infratores. A solução mais eficaz passa pela destruição da base do modelo de negócios dos *spammers*, que se baseia na idéia de que se mesmo um percentual muito reduzido de usuários comprarem os produtos anunciados por eles, seus anunciantes irão obter lucro e continuarão os financiando. A verdadeira base do problema está na conscientização dos usuários em não comprarem produtos anunciados através de *spams* que, na prática, é muito difícil de ser atingida.

Referências Bibliográficas

[1] D. M. Taveira, I. M. Moraes, M. G. Rubinstein e O. C. M. B. Duarte, *Livro Texto dos Mini-cursos do VI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, ch. Técnicas de Defesa Contra Spam, pp. 202–250. Sociedade Brasileira de Computação, 2006.

[2] Hormel Foods, "Your use of our trademark SPAM on your "Page-O-SPAM" website". http://www.rsi.com/spam/. Acessado em 27 de novembro de 2006.

[3] W. L. Cukier, S. Cody e E. J. Nesselroth, "Genres of spam: Expectations and deceptions", in *Hawaii International Conference on System Sciences (HICSS)*, pp. 1–10, janeiro de 2006.

[4] Spammer-X, J. Posluns e S. Sjouwerman, *Inside the SPAM Cartel: Trade Secrets from the Dark Side*. Syngress Publishing, 1ª ed., 2004.

[5] S. L. Pfleeger e G. Bloom, "Canning spam: Proposed solutions to unwanted email", *IEEE Security & Privacy Magazine*, vol. 3, no. 2, pp. 40–47, março de 2005.

[6] T. Emery, "MIT conference takes aim at spam emails", *Associated Press*, jan 2003.

[7] J. Klensin, *Simple Mail Transfer Protocol*. RFC 2821, abril de 2001.

[8] J. Myers, *SMTP Service Extention for Authentication*. RFC 2554, março de 1999.

[9] R. P. Laufer, I. M. Moraes, P. B. Velloso, M. D. D. Bicudo, M. E. M. Campista, D. de O. Cunha, L. H. M. K. Costa e O. C. M. B. Duarte, *Livro Texto dos Mini-cursos do V Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*,

ch. Negação de Serviço: Ataques e Contramedidas, pp. 1–63. Sociedade Brasileira de Computação, 2005.

[10] J. R. Levine, M. L. Young e R. Everett-Church, *Fighting Spam For Dummies*. John Wiley & Sons, 1ª ed., 2004.

[11] P. Gregory e M. A. Simon, *Blocking Spam & Spyware for Dummies*. Wiley Publishing, Inc., 2005.

[12] A. Ramachandran e N. Feamster, "Understanding the network-level behavior of spammers", in *ACM SIGCOMM 2006 Conference*, ACM Press, 2006.

[13] Y. Rekhter e T. Li, *A Border Gateway Protocol 4 (BGP-4)*. RFC 1771, março de 1995.

[14] J. Jung e E. Sit, "An empirical study of spam traffic and the use of DNS black lists", in *ACM SIGCOMM conference on Internet measurement (IMC' 04)*, pp. 370–375, ACM Press, 2004.

[15] Apache, "Spamassassin". http://spamassassin.apache.org/. Acessado em 27 de novembro de 2006.

[16] Apache, "Spamassassin tests performed: v3.1.x". http://spamassassin.apache.org/tests_3_1_x.html. Acessado em 27 de novembro de 2006.

[17] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.

[18] J. A. Zdziarski, "Bayesian noise reduction: Contextual symmetry logic utilizing pattern consistency analysis", *MIT Spam Conference*, janeiro de 2004.

[19] L. H. Gomes, C. Cazita, J. M. Almeida, V. Almeida e J. Wagner Meira, "Characterizing a spam traffic", in *ACM SIGCOMM conference on Internet measurement (IMC'04)*, pp. 356–369, ACM Press, 2004.

[20] J. B. Postel, *Simple Mail Transfer Protocol*. RFC 821, agosto de 1982.

[21] M. Andreolini, A. Bulgarelli, M. Colajanni e F. Mazzoni, "Honeyspam: Honeypots fighting spam at the source", in *International Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'05)*, pp. 77–83, 2005.

[22] P. O. Boykin e V. P. Roychowdhury, "Leveraging social networks to fight spam", *IEEE Computer Magazine*, vol. 38, no. 4, pp. 61–68, abril de 2005.

[23] L. H. Gomes, L. M. A. Bettencourt, V. A. F. Almeida, J. M. Almeida e F. D. O. Castro, "Quantifying social vs. antisocial behavior in email networks", *ArXiv Physics e-prints*, janeiro de 2006.

[24] M. Wong e W. Schlitt, *Sender Policy Framework (SPF) for Authorizing Use of Domains in E-MAIL, Version 1*. RFC 4408, abril de 2006.

[25] M. Wendland, "Spam king lives large off others' e-mail troubles", *Detroit Free Press*, novembro de 2002.

[26] G. Mori e J. Malik, "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 134–141, 2003.

[27] J. T. Goodman e R. Rounthwaite, "Stopping outgoing spam", in *ACM conference on Electronic commerce (EC'04)*, pp. 30–39, ACM Press, 2004.

[28] "Perl".
http://www.perl.org. Acessado em 27 de novembro de 2006.

[29] "Postfix smtp access policy delegation".
http://www.postfix.org/SMTPD_POLICY_README.html. Acessado em 27 de novembro de 2006.

[30] "Postfix".
http://www.postfix.org. Acessado em 27 de novembro de 2006.

[31] "Mail::SPF - Mail::SPF - An object-oriented implementation of Sender Policy Framework".
http://search.cpan.org/~jmehnle/Mail-SPF-2.001/lib/Mail/SPF.pm. Acessado em 27 de novembro de 2006.

[32] "Mail::Box - manage a mailbox, a folder with messages".
http://search.cpan.org/dist/Mail-Box/lib/Mail/Box.pod. Acessado em 27 de novembro de 2006.

[33] "IP::Country - fast lookup of country codes from IP addresses".
http://search.cpan.org/~nwetters/IP-Country-2.21/lib/IP/Country.pm. Acessado em 27 de novembro de 2006.

[34] "Geography::Countries - 2-letter, 3-letter, and numerical codes for countries".
http://search.cpan.org/~abigail/Geography-Countries-1.4/Countries.pm. Acessado em 27 de novembro de 2006.

[35] "Mozilla Thunderbird".
http://www.mozilla.org. Acessado em 27 de novembro de 2006.

[36] J. O. Kephart e D. M. Chess, "The vision of autonomic computing", *IEEE Computer*, vol. 36, no. 1, pp. 41–52, janeiro de 2003.

[37] T. Oda e T. White, "Developing an immunity to spam", *Genetic and Evolutionary Computation Conference (GECCO 2003)*, 2003.

[38] R. MacIntosh e D. Vinokurov, "Detection and mitigation of spam in IP telephony networks using signaling protocol analysis", *IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, pp. 49–52, abril de 2005.