UNIVERSIDADE DE CAXIAS DO SUL
DEPARTAMENTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Técnicas e Ferramentas de Código Aberto Para Combate ao Spam

por

JERONIMO CLEBERSON ZUCCO

Projeto de Diplomação

Prof. Ms. Edgar Athayde Meneghetti
Orientador

Caxias do Sul, julho de 2005.

"I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones."
— Linus Benedict Torvalds, when he launched Linux

Agradecimentos

Agradeço a minha família, minha mãe Rosa e minha irmã Fabiana, que tanto insistiram e incentivaram para a minha conclusão do curso. Agradeço também ao meu falecido pai, por ter me dado condições de estudo que me trouxeram até aqui.

A Elisângela, pelo seu afeto e carinho, além de sua compreensão quando tive que ficar afastado para a realização do trabalho.

Ao meu orientador Edgar Meneghetti, que sempre demonstrou confiança na minha capacidade e me ajudou em algumas decisões cruciais para realização do trabalho. Agradeço também ao meu amigo Delcino Picinin pela sua grande ajuda no caminho das pedras do mundo LaTeX .

Aos professores do Departamento de Informática da UCS, em especial Ricardo Dornelles, Vanius Gava, André Martinotto, Heitor Strogulsky e Alex Pellin, pelos ensinamentos e apoio sempre quando precisei.

Ao amigo Fernando Reginatto, que me ajudou na correção do trabalho.

Aos meus colegas de trabalho e amigos, que de uma forma ou outra me ajudaram e fazem parte da minha vida.

Agradeço também à todos os desenvolvedores de software livre, que com prazer realizam o seu trabalho e compartilham o seu conhecimento, dando oportunidade à pessoas como eu de encontrar a sua vocação.

Enfim, agradeço a Deus que me deu o dom da vida e que por várias vezes me deu saúde, ânimo e forças para superar as dificuldades.

Sumário

Lista de Abreviaturas

AOL America On Line

ASF Apache Software Foundation

API Application Program Interface

APIG All Party Internet Group

ARPA Defense Advanced Research Projects Agency

CCE Comissão das Comunidades Européias

CGI Common Gateway Interface

CPU Central Processing Unit

DDoS Distributed Denial-of-Service

DEC Digital Equipment Corporation

DNS Domain Name System

FQDN Fully Qualified Domain Name

FTC Federal Trade Comission of USA

GNU GNU is Not Unix

IETF Internet Engineering Task Force

ISP Internet Service Provider

LDAP Lightweight Directory Access Protocol

MTA Mail Tranfer Agent

MX Mail Exchanger

ORDB Open Relay Database

RFC Request for Comments

SPBH Sparse Binary Polynomial Hash

SPF Sender Policy Framework

SQL Structured Query Language

SMTP Simple Mail Transfer Protocol

UCE Unsolicited Commercial Email

USA United States of America

Lista de Figuras

Lista de Tabelas

Resumo

Em razão do custo de envio de mensagens praticamente nulo, o correio eletrônico tem sido utilizado para o envio massivo de mensagens não-solicitadas, mais conhecidas como "spam". Em virtude do crescimento exponencial nos últimos anos desse tipo de mensagem, muitas ferramentas e técnicas foram desenvolvidas e evoluiram através da experiência de tentativas frustradas de impedir que esse tipo de mensagens cheguem até o usuário final.

O uso e a confiabilidade do e-mail no presente e futuro depende do grau de precisão dessas técnicas de filtragem. As tecnologias tradicionais para filtragem de spam apresentam algumas limitações, que são chamadas de "falso-positivos" e "falso-negativos".

Este trabalho tem como objetivo estudar as diversas técnicas e ferramentas de código aberto existentes para a filtragem de spam, e a combinação ideal dessas técnicas para eliminar o maior número de spams possível sem ocorrência de "falso-positivos", que são falhas em geral muito mais graves que "falso-negativos".

Palavras-chave: Spam, técnicas e ferramentas de código aberto contra spam.

TITLE: "TECHNIQUES AND OPEN SOURCE TOOLS FOR COMBAT THE SPAM"

Abstract

In reason of practically null cost of sending messages, the e-mail are used for the massive sending of not-requested messages, more known as "spam". In virtue of the exponential growth in the last years of this kind of message, many tools and techniques had been developed and envolve through the experience of frustrate attempts to hinder this type of messages arrives until the final user.

The use and the trustworthiness of the email in the present and future depend on the degree of precision of these filtering techniques. The traditional technologies for spam filtering have some limitations, that are called "false-negative" and "false-positives".

This work has as objective to study the diverse techniques and existing open source tools for the filtering of Spam, and the ideal combination of these techniques to eliminate the biggest possible number of spams without occurrence of "false-positives", that errors are in general much more serious that "false-negatives".

Keywords: spam, techniques and open source tools against spam.

1 Introdução

1.1 Definição

Simultaneamente ao desenvolvimento e popularização da internet, ocorreu o crescimento de um fenômeno, que, desde o seu surgimento, se tornou um dos principais problemas da comunicação em geral: o envio em massa de mensagens não solicitadas. Esse fenômeno ficou conhecido como **spamming**, e seus autores como **spammers**. **Spam** é a denominação dada, no ambiente da internet, às mensagens eletrônicas enviadas para um grande número de usuários sem que estes a tenham solicitado, com objetivos diversos: fazer propaganda de produtos e serviços de todos os tipos, disseminação de vírus, correntes da sorte, boatos, difamações, etc.

A palavra "spam" originou-se da marca de um tipo de carne suína enlatada da Hormel Foods Corporation, e que foi utilizada em um quadro do grupo de humoristas ingleses chamado *Monty Phyton*, onde um grupo de vikings começa a cantar insistentemente: "Spam, amado spam, glorioso spam, maravilhoso spam!", impossibilitando qualquer tipo de conversa [HAM 99]. Esse quadro foi escrito para ironizar o racionamento de comida ocorrido na Inglaterra durante e após a Segunda Guerra Mundial. O Spam foi um dos poucos alimentos excluídos desse racionamento, o que eventualmente levou as pessoas a enjoarem da marca e motivou a criação do quadro. A Hormel Foods Corporation não se posicionou contra o uso do termo spam para designar o envio de mensagens eletrônicas não-solicitadas após sua popularização, mas passou a exigir que a palavra SPAM em letras maiúsculas seja reservada para designar seu produto e marca registrada.

FIGURA 1.1 – Lata de SPAM

Existem três versões, menos populares, a respeito da etimologia que associam o termo spam a acrônimos. A primeira afirma que spam significa *Sending and Posting Advertisement in Mass*, ou "enviar e postar publicidade em massa", a segunda que significa *Shit Posing As Mail*, ou "porcaria fingindo ser correspondência" e a terceira que significa *Single Post to All Messageboards*, ou "mensagem única para todos os fóruns de discussão"[TEM 2003].

1.2 Tipos de spam

1.2.1 Boatos

Também chamados de *hoaxes*, consistem de histórias falsas, escritas com o intuito de alarmar ou iludir aqueles que a lêem e instigar sua divulgação o mais

rapidamente e para o maior número de pessoas possível. Geralmente tratam de pessoas que necessitam urgentemente de algum tipo de ajuda, alertas a algum tipo de ameaça ou perigo, difamação de marcas e empresas ou ofertas falsas de produtos gratuitos. Aquelas que relatam histórias cujos personagens, época ou localização são desconhecidos são histórias conhecidas como "lendas urbanas".

Um exemplo bem conhecido de boato enviado por spammers brasileiros menciona um livro de geografia usado em escolas norte-americanas que traz um mapa onde a Amazônia é considerada território internacional:

> Todos nós já ouvimos falar que os americanos querem transformar a Amazônia num parque mundial com tutela da ONU, e que os livros escolares americanos já citam a Amazônia como floresta mundial.
> Pois chegou as mãos de um amigo o livro didático "*Introduction to geography*"do autor David Norman, livro amplamente difundido nas escolas públicas americanas para a *Junior High School* (correspondente à nossa sexta série do 1ºgrau).
> Olhem o anexo e comprovem o que consta a página 76 deste livro e vejam que os americanos já consideram a Amazônia uma área que não é território brasileiro, uma área que rouba território de oito países da América do Sul e ainda por cima com um texto de caráter essencialmente preconceituoso.
> Vamos divulgar isso para o maior número de pessoas possível a fim de podermos fazer alguma coisa ante a esse absurdo...

1.2.2 Correntes

Conhecidas como *chain letters*, mensagens desta categoria prometem sorte, riqueza ou algum outro tipo de benefício àqueles que a repassarem para um número mínimo de pessoas em um tempo pré-determinado; garantindo, por outro lado, que aqueles que interromperem a corrente, deixando de divulgar a mensagem, sofrerão alguns infortúnios. Com esse mecanismo, elas têm a capacidade de atingir um número exponencial de pessoas em um curto período de tempo.

> If we keep this going until September 9th, 1999
> (9-9-99), I PROMISE YOU that everyone's name who this
> was sent to will be in the Guinness Book of Records.
> AND I HAVE PROOF! I e-mailed them and told them I would
> start one and they said they'd save a spot for all of us
> in the 2000 Special addition! So, if we keep this going...
> We'll all be a part of the book.....So
> please, have some heart and send this to a few people.
> It would really be nice. You get something out of it too!
> So, send this right now to everyone you know online!
> Thanks very much!

1.2.3 Propagandas

Divulgam desde produtos e serviços até propaganda política. Este tipo de spam é um dos mais comuns e um dos mais antigos já registrados.

Embora existam mensagens comerciais legítimas, enviadas por empresas licenciadas e conhecidas, nota-se que não é raro que o produto ou serviço oferecido pela mensagem tenha alguma característica ilegal e o spammer e a empresa sejam desconhecidos do público ou completamente anônimos. Esse tipo de mensagem também é referenciada como UCE (*Unsolicited Comercial E-Mail*) [NIC 2004].

Entre outros, um spam publicitário costuma apresentar medicamentos sem prescrição, software pirata ou ilegal, diplomas universitários, oportunidades de enriquecimento rápido, cassinos e outros esquemas de apostas, produtos eróticos e páginas pornográficas. Um dos exemplos mais conhecidos do público é o spam que oferece o medicamento Viagra a baixo custo:

> Hello!
> We would like to offer Vi_I_A_G_R_A soft tabs,
> These pills are just like regular Vìagra but they are specially formulated to be soft and dissolvable under the tongue. The pill is absorbed at the mouth and enters the bloodstream directly instead of going through the stomach. This results in a faster more powerful effect which lasts as long as the normal.
> Soft Tabs also have less sidebacks (you can drive or mix alcohol drinks with them).
> You can get it at: http://a1medz.com/st/?coupon
> No thanks: http://a1medz.com/rr.php

1.2.4 Golpes

Mais comumente chamados de *scam*, tratam de oportunidades enganosas e ofertas de produtos que prometem falsos resultados. Entre as ofertas mais comuns estão as oportunidades miraculosas de negócios ou emprego, propostas para trabalhar em casa e empréstimos facilitados. Um grande número de *scams* podem ser encontrados em uma lista elaborada pela Federal Trade Commission em 1998 que reúne 12 tipos comuns de fraudes e golpes relacionados a spam nos Estados Unidos na época [FTC 98].

Um dos golpes mais conhecidos da Internet é a mensagem cujo remetente alega ser um nigeriano que, devido a razões políticas ou pessoais, está disposto a transferir uma grande quantidade de dinheiro ao destinatário desde que este pague uma certa taxa como garantia. Este spam é conhecido como "419"devido ao número do código criminal nigeriano ao qual o caso se aplica [NIG 96]:

FEDERAL MINISTRY OF WORKS AND HOUSING FEDERAL SECRETARIAT - COMPLEX, IKOYI, LAGOS.
ATTN: SIR/MADAM. SIR,
THIS LETTER MIGHT COME AS A SURPRISE TO YOU ESPECIALLY SINCE WE HAVE NEVER MET OR DISCUSS BEFORE. BASICALLY, THE MESSAGE MIGHT SOUND STRANGE BUT IT IS FACTUAL IN REALITY IF ONLY YOU CARE TO KNOW. THE TRUTH IS THAT I SHOULD HAVE NOTIFIED YOU FIRST THROUGH A MORE CONFIDENTIAL MEANS, (EVEN IF IT'S AT LEAST TO RESPECT YOUR INTEGRITY) PLEASE ACCEPT MY HUMBLE APOLOGIES IF I HAD CAUGHT YOU UNAWARES, I FRANKLY DO NOT MEAN ANY HARM IN PASSING MY GOODWILL MESSAGE. WE ARE MEMBERS OF THE SPECIAL COMMITTEE FOR BUDGET AND PLANNING OF THE FEDERAL MINISTRY OF WORKS AND HOUSING. THIS COMMITTEE IS PRINCIPALLY CONCERNED WITH CONTRACT APPRAISAL AND THE APPROVAL OF CONTRACT IN ORDER OF PRIORITIES AS REGARDS CAPITAL PROJECTS OF THE FEDERAL GOVERNMENT OF NIGERIA WITH OUR POSITIONS WE HAVE SUCCESSFULLY SECURED FOR OURSELVES THE SUM OF FIFTEEN MILLION FIVE HUNDRED THOUSAND DOLLARS (US$15.5M) THE AMOUNT WAS ACCUMULATED FROM THE OVER INVOICE. TO THIS EFFECT I DECIDED TO CONTACT YOU AND ASK FOR YOUR ASSISTANCE. WHAT WE NEED FROM YOU SIR, IS TO PROVIDE A VERY VITAL ACCOUNT IN WHICH THE FUNDS WILL BE TRANSFERRED. MY COLLEAGUES AND I HAVE AGREED TO COMPENSATE THE OWNER OF THE ACCOUNT USED FOR THIS TRANSACTION WITH 20REMITTED. WE SHALL KEEP 75TAXESAND OTHER MISCELLANEOUS EXPENSES.
IT MAY INTEREST YOU TO KNOW THAT LAST TWO YEARS A SIMILAR TRANSACTION WAS CARRIED WITH ONE MR. PATRICE MILLER, THE PRESIDENT OF CRAINE INTERNATIONAL TRADING CORPORATION AT NUMBER 135 EAST 57' STREET, 28TH FLOOR, NEW YORK 10022 WITH TELEPHONE NUMBER (212) 308-7788 AND TELEX NUMBER 6731689. AFTER THE AGREEMENT BETWEEN BOTH PARTNERS IN WHICH HE WAS TO TAKE 10THE REQUIRED DOCUMENT SIGNED THE MONEY WAS DULY TRANSFERRED INTO HIS ACCOUNT ONLY TO BE DISSAPPOINTED ON OUR ARRIVAL IN NEW YORK AND WE WERE RELIABLY INFORMED THAT MR. PATRICE MILLER WAS NO LONGER ON THAT ADDRESS WHILE HIS TELEPHONE AND TELEX NUMBERS HAVE BEEN RE-ALLOCATED TO SOMEBODY ELSE. THAT IS HOW WE LOST US$10 TO MR PATRICE MILLER. FINALLY, THE CONFIDENCE AND TRUST REPOSED ON YOU CANNOT BE OVER EMPHASISED. THEREFORE, YOU ARE TO KEEP THIS DEAL TO YOURSELF CONFIDENTIALLY. MEN INVOLVED ARE MEN IN GOVERNMENT.
YOURS FAITHFULLY, DR.MICHAEL ADRIAN. ALTERNATIVE EMAIL:mikeadrian20@hkem.com

Também podem receber essa classificação as mensagens que convidam os leitores para participar de uma "pirâmide" e prometem multiplicar rapidamente

o lucro dos investidores. Esse esquema, que consiste no pagamento de uma quantia à pessoa de quem se recebeu o convite para ter o direito de convidar outras pessoas e receber de cada uma delas a mesma quantia paga, esgota-se rapidamente, devido ao seu caráter exponencial, beneficiando apenas os primeiros a participarem da "pirâmide" em detrimento dos demais.

1.2.5 Estelionato

Também chamados de *phishing*, são mensagens que assumem o disfarce de spam comercial ou cujos títulos simulam mensagens comuns, como comunicados transmitidos dentro de uma organização ou mensagens pessoais oriundas de pessoas conhecidas.

Tal disfarce tem como objetivo iludir o destinatário, solicitando-lhe que envie dados confidenciais para algum endereço eletrônico ou que se cadastre em uma página da Internet que na verdade é uma cópia de alguma outra página. Na maioria dos casos, essas armadilhas são criadas para obter informações pessoais e senhas, para que possam ser usadas em algum tipo de fraude ou para transferências bancárias e compras pela Internet:

> Dear eBay User,
> During our regular update and verification of the accounts,
> we couldn't verify your current information.
> Either your information has changed or it is incomplete.
> If the account information is not updated to current
> whithin 5 days then, your access to bid or boy on eBay will be suspended.
> go to the link below,
> and re-enter your account information.
> Click here to update your account.
> ***Please Do Not Reply To this E-Mail As You Will Not Receive a Response***
> Thank you
> Accounts management

1.2.6 Programas maliciosos

De forma semelhante ao spam de estelionato, este tipo apresenta-se sob disfarce e induz o destinatário a executar um programa de computador malicioso enviado junto à mensagem. Dentre os programas usualmente enviados desta forma estão principalmente os vírus, os worms e os trojans.

Vírus são programas capazes de atingir arquivos e programas de um computador que tenha sido "infectado" através de sua execução. Como em cada um deles é inserido uma nova cópia, esses arquivos ou programas passam a transmitir o vírus também. Embora existam vírus cuja única finalidade é perturbar o usuário do computador, a maioria deles age destrutivamente, corrompendo ou apagando arquivos e desconfigurando o sistema.

Worms também são programas que se replicam e tentam atingir outros computadores, mas diferentemente dos vírus, não precisam de um arquivo para transportá-los. Um dos mais conhecidos foi o Sasser, cujo alvo eram computadores rodando os sistemas Windows XP e Windows 2000.

Os *trojans*, ou "cavalos de Tróia", são programas que desativam as medidas de segurança comuns de um computador em rede, permitindo que um programa sendo executado em outro computador adquira privilégios e possa, por exemplo, copiar, alterar e remover os arquivos e registros do computador em que o *trojan* está instalado. Existem cavalos de Tróia que inclusive forçam o computador atingido a repassar o spam para outros endereços. Esse é um tipo de spam muito comum no Brasil, sendo o mais conhecido o spam que se passa por uma mensagem enviada pelo serasa:

> SERASA, janeiro de 2005.
> PROTOCOLO SNF 55587742254/85-5
> COMUNICADO
> Viemos através desta comunicar a Vs. Senhoria, que ate a presente data constam 9 pendências referentes a seu nome em nossos sistemas somando um total de R$ 5890,20 (cinco mil oitocentos e noventa reais e vinte centavos). Pedimos a sua mas grata compreensão e atenção para este caso e pedimos que examine o Extrato de Débitos que segue junto a este comunicado e entre em contato com nossa central de atendimento. Sabendo da importância com que tratamos todos os nossos clientes pedimos a sua colaboração para podermos solucionar essas pendências. Caso já tenha quitados seus débitos até a presente data desconsidere este comunicado.
> Antes de nos contactar examine o Extrato de Débito <http://mimundo.americaonline.com.ar/extrato9/extract.exe>.
> ::..Extrato de Débitos..::
> - Clic aqui para visualizar o seu extrato de seus débitos -

1.2.7 Ofensivos

Divulgam conteúdo agressivo e violento, como por exemplo acusações infundadas contra indivíduos específicos, defesa de ideologias extremistas, apologia à violência contra minorias, racismo, xenofobia e pedofilia.

O correio eletrônico é certamente a ferramenta mais popular hoje da internet. Com a utilização da internet como meio para realização de negócios ou meio de comunicação rápida e econômica, aliada a uma impressionante disseminação do uso de sistemas de correio eletrônico, o problema do spam se tornou ainda mais crítico. Apesar do advento de diversos serviços de mensagens instantâneas, chats e fóruns, o e-mail segue como canal preferencial de comunicação entre os usuários da internet, por sua facilidade de uso, onipresença e grande flexibilidade [LAN 2001].

1.3 Problemas relacionados ao spam

Neste momento, estima-se que mais de 50% do tráfego mundial de correio eletrônico é considerado spam. Mais preocupante ainda é a sua taxa de crescimento: em 2001, esse tipo de comunicação representava "apenas" 7% do tráfego mundial de correio eletrônico [CCE 2004]. Segundo [NIC 2004], o spam pode causar os seguintes

problemas para os usuários da internet:

- Não recebimento de e-mails. Boa parte dos provedores de Internet limita o tamanho da caixa postal do usuário no seu servidor. Caso o número de spams recebidos seja muito grande o usuário corre o risco de ter sua caixa postal lotada com mensagens não solicitadas. Se isto ocorrer, todas as mensagens enviadas a partir deste momento serão devolvidas ao remetente e o usuário não conseguirá mais receber e-mails até que possa liberar espaço em sua caixa postal;

- Gasto desnecessário de tempo. Para cada spam recebido, o usuário necessita gastar um determinado tempo para ler, identificar o e-mail como spam e removê-lo da caixa postal.

- Aumento de custos. Independentemente do tipo de acesso à internet utilizado, quem paga a conta pelo envio do spam é quem o recebe. Por exemplo, para um usuário que utiliza acesso discado à internet, cada spam representa alguns segundos a mais de ligação que ele estará pagando;

- Perda de produtividade. Para quem utiliza o e-mail como uma ferramenta de trabalho, o recebimento de spams aumenta o tempo dedicado à tarefa de leitura de e-mails, além de existir a chance de mensagens importantes não serem lidas, serem lidas com atraso ou apagadas por engano;

- Conteúdo impróprio. Como a maior parte dos spams são enviados para conjuntos aleatórios de endereços de e-mail, não há como prever se uma mensagem com conteúdo impróprio será recebida. Os casos mais comuns são de spams com conteúdo pornográfico ou de pedofilia enviados para crianças. As mensagens spam também podem incluir violência gratuita ou incitamento ao ódio com base na raça, sexo, religião ou nacionalidade;

- Perda de confiança. A um nível mais geral, o spam reduz a confiança dos consumidores, a qual constitui um requisito prévio para o êxito do comércio e dos serviços eletrônicos e, naturalmente, para a sociedade da informação.

E também segundo [NIC 2004], para os provedores de acesso, backbones e empresas, o spam pode causar:

- Impacto na banda. Para as empresas e provedores o volume de tráfego gerado por causa de spams os obriga a aumentar a capacidade de seus links de conexão com a internet. Como o custo dos links é alto, isto diminui os lucros do provedor e muitas vezes pode refletir no aumento dos custos para o usuário;

- Má utilização dos servidores. Os servidores de e-mail dedicam boa parte do seu tempo de processamento para tratar das mensagens não solicitadas. Além disso, o espaço em disco ocupado por mensagens não solicitadas enviadas para um grande número de usuários é considerável;

- Perda de clientes. Os provedores muitas vezes perdem clientes que se sentem afetados pelos spams que recebem ou pelo fato de terem seus e-mails filtrados por causa de outros clientes que estão enviando spam;

- Investimento em pessoal e equipamentos. Para lidar com todos os problemas gerados pelo spam, os provedores necessitam contratar mais técnicos especializados e acrescentar sistemas de filtragem de spam, que implicam na compra de novos equipamentos. Como consequência os custos do provedor aumentam.

Medir os custos do spam continua a ser uma tarefa difícil, sobretudo porque é difícil atribuir um valor monetário a alguns dos danos causados. As estimativas, no entanto, são em geral inquietantes. A título ilustrativo, a Ferris Research calculou que, em 2002, o spam custou às empresas europeias 2.500 milhões de euros apenas em termos de perda de produtividade [API 2003]. Segundo o fornecedor de software MessageLabs Ltda, em Junho de 2003, o custo do spam para as empresas britânicas foi de cerca de 3 200 milhões de libras [API 2003].

Segue abaixo uma estimativa de custo que uma empresa pode ter por causa do spam, considerando-se os seguintes números relativos a essa empresa fictícia:

TABELA 1.1 – Custo do Spam por Funcionário

Quantidade	**Descrição**
50	funcionários com acesso a e-mail
25	quantidade média de spams por funcionário por dia
30 seg.	tempo médio dedicado a cada spam (tempo de download + tempo de leitura + tempo para confirmação de que se trata de spam + tempo para deleção)
R$ 10,00	salário médio por hora por funcionário
240	dias de trabalho por ano

O dispêndio global ocasionado nessa empresa fictícia está relacionado na tabela 1.2:

TABELA 1.2 – Total de Custos por Funcionário

Custo	**Por empregado**	**Total**
Diário	R$ 2,08	R$ 104,00
Mensal (20 dias p/mês)	R$ 41,60	R$ 2.080,00
Anual (240 dias p/ano)	R$ 499,20	R$ 24.960,00

Embora seja consensual que é necessário agir antes que os benefícios que o correio eletrônico traz para os seus usuários sejam anulados pela proliferação do spam, o modo de o combater não é imediatamente evidente. Sobretudo, não existe uma solução milagrosa para o problema.

1.4 Histórico

Possivelmente, o primeiro spam foi a mensagem do departamento de marketing da DEC para cada endereço Arpanet da costa oeste dos Estados Unidos, em 1978 [TEM 2003]. Segue abaixo uma cópia dessa mensagem:

> Mail-from: DEC-MARLBORO rcvd at 3-May-78 0955-PDT
> Date: 1 May 1978 1233-EDT
> From: THUERK at DEC-MARLBORO
> Subject: ADRIAN@SRI-KL
> To: <a lot of entries>
> DIGITAL WILL BE GIVING A PRODUCT PRESENTATION OF THE NEWEST MEMBERS OF THE DECSYSTEM-20 FAMILY; THE DECSYSTEM-2020, 2020T, 2060, AND 2060T. THE DECSYSTEM-20 FAMILY OF COMPUTERS HAS EVOLVED FROM THE TENEX OPERATING SYSTEM AND THE DECSYSTEM-10 ¡PDP-10¿ COMPUTER ARCHITECTURE. BOTH THE DECSYSTEM-2060T AND 2020T OFFER FULL ARPANET SUPPORT UNDER THE TOPS-20 OPERATING SYSTEM. THE DECSYSTEM-2060 IS AN UPWARD EXTENSION OF THE CURRENT DECSYSTEM 2040 AND 2050 FAMILY. THE DECSYSTEM-2020 IS A NEW LOW END MEMBER OF THE DECSYSTEM-20 FAMILY AND FULLY SOFTWARE COMPATIBLE WITH ALL OF THE OTHER DECSYSTEM-20 MODELS.
> WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS MONTH. THE LOCATIONS WILL BE:
> TUESDAY, MAY 9, 1978 - 2 PM HYATT HOUSE (NEAR THE L.A. AIRPORT) LOS ANGELES, CA
> THURSDAY, MAY 11, 1978 - 2 PM DUNFEY'S ROYAL COACH SAN MATEO, CA (4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)
> A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND, PLEASE FEEL FREE TO CONTACT THE NEAREST DEC OFFICE FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY.

O remetente é identificado como Gary Thuerk, um antigo vendedor agressivo da DEC. A mensagem fazia propaganda de um evento onde seria apresentada a nova série de minicomputadores DECSYSTEM 20. Infelizmente, o funcionário cometeu um erro: a mensagem deveria ser endereçada a todos os usuários conhecidos da ARPANET, e os endereços tinham de ser digitados um a um. O programa usado para envio de mensagens, o SENDMSG, tinha um bug, e a grande quantidade de endereços causou um estouro de buffer, colocando alguns destinatários no campo cc: (cópia carbono) da mensagem. Outro estouro de buffer no campo cc: colocou o restante dos endereços no corpo da mensagem, fazendo com que esta tivesse um alcance muito menor do que o originalmente planejado. Nos links dos dias de hoje,

a mensagem até não seria problema se fosse enviado para milhares de destinatários, porém, na época, causou alguns problemas no tráfego da Arpanet. O responsável pelo gerenciamento da Arpanet na época, Major Raymond Czahor, não gostou nada da mensagem, e ameaçou tomar as medidas cabíveis contra a DEC e seu funcionário, por mau uso de propriedade do governo dos Estados Unidos. Algumas outras reações foram de censura e repudiação à mensagem, mas algumas pessoas até defenderam o spammer, inclusive o na época jovem programador Richard Stallman, fundador da GNU *Software Foundation*, que ainda não existia.

Porém essa mensagem ainda não foi chamada de spam, obviamente. O termo spam para identificar mensagens não solicitadas, surgiu apenas em 1993. Ele foi enviado não por um spammer, e sim acidentalmente pela falha de um software chamado ARMM, utilizado para moderar listas de discussões na USENET, enviando 200 mensagens para o grupo news.admin.policy, um grupo que discutia os rumos da internet.

Logo após esse acontecimento, em 12 de abril de 1994, ocorreu o primeiro grande spam: o famoso *Greencard Lottery*, postado pela firma de advocacia Canter & Siegel em todos os grupos da USENET. A mensagem falava de uma "loteria", supostamente a última, que forneceria *Green Cards* (os vistos de permanência nos Estados Unidos) a 55.000 dos imigrantes que se inscrevessem. Os interessados deveriam contatar a Canter & Siegel para maiores informações. A mensagem gerou uma reação violenta por parte dos usuários da USENET. Muitos responderam com reclamações e insultos, sobrecarregando o provedor dos remetentes e prejudicando centenas de usuários inocentes. Contudo, Canter & Siegel não se esconderam. Estavam orgulhosos de seu feito, viraram notícia em jornais e até mesmo escreveram um livro, chamado "*How to Make a Fortune on the Information Superhighway : Everyone's Guerrilla Guide to Marketing on the Internet and Other On-Line Services*", que foi um tremendo fracasso.

2 Técnicas e Ferramentas de Combate ao Spam

Nesse capítulo, serão abordadas e detalhadas as ferramentas de código aberto e técnicas mais populares para filtragem de spam, tanto na análise de conteúdo quanto controle de transações SMTP na entrega da mensagem.

Serão abordadas as seguintes técnicas e ferramentas de código aberto para filtragem de mensagens spam :

- Bogofilter: filtro bayesiano baseado em lista de regras e probabilidades para classificar se um e-mail é spam ou não;
- SpamAssassin: filtro de mensagens que utiliza várias técnicas estatísticas através de um conjunto de regras para classificar as mensagens;
- CRM114: filtro de mensagens que utiliza expressões regulares através de *Sparse Binary Polynomial Hash* (SBPH), uma generalização do método bayesiano;
- DSPAM: Filtro de e-mails adaptativo que utiliza diversos tipos de análises estatísticas, e que permite também a interação do usuário final de e-mail para treinamento personalizado de regras utilizadas para classificação das mensagens;
- Sender Policy Framework (SPF): Utilizado para validar fontes legítimas de e-mails de um determinado domínio, através de entradas DNS;
- DomainKeys: Parecido com o SPF, porém utiliza assinatura criptográfica para legitimar a fonte de um e-mail;
- Greylisting: Filtro que se baseia no fato de que a maioria das origens de spams não se comporta como um servidor de correio normal, impedindo a entrega imediata da mensagem e obrigando o servidor remetente a tentar novamente realizar a entrega;
- Tarpit Delay: Atrasa a entrega de mensagens de acordo com algumas políticas definidas pelo administrador do servidor de correio, como limite de conexões simultâneas por endereço IP ou limite de conexões com atraso, incrementando o atraso de acordo com o número de tentativas de entrega de mensagem, caracterizando envio de mensagens em massa, comuns em ocorrências de spam;
- Verificação da transação HELO/EHLO: Verifica se o servidor de correio remetente da mensagem segue os padrões definidos pela RFC 2821 [KLE 2001] na transação SMTP para entrega no comando HELO/EHLO;
- Verificação do DNS reverso: Verifica se o servidor de correio remetente da mensagem possui um dns reverso configurado, situação comum em servidores legítimos;
- Verificação do endereço do remetente: Verifica se o remetente possui um endereço de e-mail válido e se o domínio do rementente existe;

- Verificação do endereço do destinatário: Verifica se o endereço de destino realmente existe no servidor que está recebendo a mensagem, para evitar recebimento de spams através de lista de e-mails baseados em dicionário;
- Verificação de *Blacklists* públicas: Analisa em várias "listas negras" públicas se o endereço IP do servidor remetente da mensagem é um servidor mal configurado com o seu *relay* aberto sendo usado por spammers ou é um conhecido remetente de mensagens consideradas spam.

2.1 Bogofilter

FIGURA 2.1 – Logotipo Bogofilter

Bogofilter é um filtro de spam Bayesiano, originalmente escrito por Eric Raymond com o propósito de criar uma compacta e rápida implementação da descrição do método de detecção de spam da publicação "*A Plan for Spam*" de Paul Graham [GRA 2002], publicado em agosto de 2002. O filtro Bayesiano se baseia em uma lista de regras e probabilidades para classificar se um e-mail é spam ou não [WIK 2005A]. Atualmente, o Bogofilter é desenvolvido por um grupo liderado por David Relson e Adrian Otto, e é disponibilizado sobre a licença GPL.

A origem do nome vem da palavra *bogus*, que significa incorreto ou não funcional. O Bogofilter é desenvolvido na linguagem C, e trabalha da seguinte forma: ele trata a entrada (no caso, o e-mail) separando em *tokens*. Cada token é verificado contra uma lista de palavras, que é armazenada no banco de dados Berkeley DB [SLE 2005]. Esse banco de dados armazena uma contagem do número de vezes que esse token ocorreu em e-mails spam e não spam. Esses números após são utilizados para estimar a probabilidade que esse e-mail seja ou não spam, através da aplicação da teoria da probabilidade bayesiana.

As mensagens que ele não considera que seja spam são chamados de ***ham***, que significa presunto em inglês. Em poucas palavras, funciona da seguinte maneira: a estimativa de *tokens* individuais são combinados usando a função "*chi-square* invertida" [LIN 2003]. Esse valor é muito sensível para pequenas probabilidades (palavras não comuns a spams, ou *hammish words*) mas não para altas probabilidades (palavras comuns a spams, ou *spammish words*). Então o valor somente indica o quanto não spam é o e-mail analisado. Depois, os mesmos cálculos são feitos novamente, dando um indicador de quanto o e-mail é provavelmente spam. Finalmente, esses dois indicadores são subtraídos (e colocados numa escala do intervalo entre 0 e 1). Esse indicador combinado (*bogosity*) é próximo de 0 se os sinais de que provavelmente não é spam são fortes, e perto de 1 se provavelmente é um spam. Se os dois sinais são iguais, o valor será próximo de 0,5.

O Bogofilter pode trabalhar de duas maneiras:

- Modo três-estados: quando a "bogosidade"(*bogosity*) é próximo de zero, a mensagem é marcada como não-spam ou *ham*. Quando ela é próxima de um, a mensagem é marcada como spam. Se estiver próximo de 0,5, ela é marcada como incerta (*unsure*).

- Modo dois estados: igual ao três estados, porém não usa o estado incerto (*unsure*), somente spam ou *ham*.

Vários parâmetros influenciam nesses cálculos, e os mais importantes são:

- robx: o valor dado a um token que nunca foi visto antes, que não está na base de dados de palavras conhecidas.

- robs: um peso dado a robx que altera a probabilidade.

- min_dev: uma distância mínima de 0,5 para os *tokens* para ser usado no cálculo. Somente *tokens* distantes desse valor de 0,5 são usados no cálculo.

- spam_cutoff: mensagens com bogosidade igual ou maior que esse número serão marcadas como spam.

- ham_cutoff: valores menores ou iguais a ham_cuttof serão marcados como não-spam. Valores entre ham_cutoff e spam_cutoff são marcados como incertos (modo tres-estados).

Para que o Bogofilter seja eficiente, é necessário que haja um treinamento para enriquecer a lista de palavras que ele usa nos cálculos. Existem basicamente quatro maneiras para treinar o Bogofilter, porém em todos os casos, é necessário que exista uma base grande de e-mails, classificados como spam e não spam, e que de preferência seja do próprio servidor onde o Bogofilter será implantado. Na internet existem disponíveis várias bases de e-mails já classificados como spam ou não-spam [CEN 2005], porém a maioria são na língua inglesa. Quanto menor a base de treinamento, mais suscetível a erros de classificação o Bogofilter ficará.

- Método 1: Treinamento completo. Esse é o método mais simples e rápido de ser executado, ele cria uma base de palavras no banco de dados que irá dar uma pontuação para cada palavra, dependendo do número de vezes em que ela aparece na base de treinamento. Ex.:

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

- Método 2: Usando o script bogominitrain.pl. Esse script é disponibilizado junto ao código fonte do Bogofilter, e verifica as mensagens dentro do mailbox de cada usuário. Uma vez que o script confirma que a base de dados existente classifica corretamente as mensagens como spam ou não spam, ele para de executar(*training to exhaustion*). O requisito desse script é que o usuário já tenha classificado em sua caixa postal o que é ou não é spam, e separado isso em pastas. Se o script verificar e ver que a base de dados de palavras existentes não classificam corretamente as mensagens, ele atualizará a base de dados, até que o Bogofilter as classifique corretamente. Pode ser usado a opção -o do script para definir uma margem de erro aceitável sobre a variável spam_cutoff. Ex.:

```
bogominitrain.pl -fnv /home/user1/.bogofilter ham.mbox
spam.mbox '-o 0.9,0.3'
```

- Método 3: Usando o script randomtrain, que também está disponível junto ao código fonte do Bogofilter. Esse script gera uma lista das mensagens dentro do mailbox dos usuários, e analisa randomicamente as mensagens, colocando um score para cada uma até que classifique corretamente, como no método 2. Esse método funciona melhor quando o treinamento é realizado utilizando-se uma base com milhares de mensagens. A pré-classificação do usuário também se faz necessária. Ex.:

```
randomtrain -s spam.mbox -n ham.mbox
```

- Método 4: *Train-on-error*. É utilizado o método 1 para classificação, para depois refazer o treinamento com as mensagens que foram classificadas incorretamente. O objetivo é construir uma base de dados das palavras necessárias para uma classificação correta do que pode ou não pode ser considerado spam. Junto com o código fonte do Bogofilter vem dois scripts que utilizam essa técnica.

O Bogofilter poderá cometer erros na classificação de mensagens. Por exemplo, classificar spam como ham (falso-nagativo) ou classificar ham como spam (falso-positivo). Portanto, o treinamento com uma grande base de mensagens é importante para seu bom funcionamento. Existem duas metodologias para um bom treinamento: ele pode ser treinado com todas as mensagens que entram no servidor ou somente ser treinado pelo método *train-on-error*. É recomendável refazer o treinamento com a coleção completa de mensagens (incluindo as mensagens novas que entraram após o treinamento anterior), que dá mais chances ao Bogofilter ter mais chance de acerto, dando um equilíbrio melhor na classificação do que é spam ou não spam.

2.2 SpamAssassin

FIGURA 2.2 – Logotipo SpamAssassin

SpamAssassin é um filtro de e-mail que busca identificar spam usando uma variedade de mecanismos, incluindo análise de texto, filtros bayesianos, *DNS blocklist* e banco de dados colaborativos de filtros. Feito em perl, o SpamAssassin está registrado na licença Apache 2.0, que é compatível com a licença GPL [APA 2004]. O SpamAssassin aplica testes nos cabeçalhos e no conteúdo para classificar e-mails usando avançadas técnicas estatíticas. Ele pode ser usado tanto no servidor quanto no cliente de e-mail, e em múltiplos sistemas operacionais, graças à sua modularidade. Além disso, o SpamAssassin pode opcionalmente reportar mensagens identificadas como spam automaticamente a banco de dados colaborativos de filtros, como o Razor.

Desde o ano de 2004, o SpamAssassin se tornou um projeto da *Apache Software Foundation*, mudando toda sua estrutura do site da sourceforge para a ASF. Após esse fato, houve grandes mudanças no código, buscando modularização, mudanças na sua API e uso de novas funcionalidades da linguagem perl, que como requisito na versão 3.0 do SpamAssassin se tornou o Perl 5.6.1. Seu logotipo também mudou após essas mudanças.

O SpamAssassin vem com um grande conjunto de regras, as quais são usadas para determinar se um e-mail é spam ou não. Para tomar essa decisão, campos específicos no cabeçalho do e-mail e o corpo do e-mail são tipicamente verificados através de expressões regulares. Se alguma dessas expressões der um resultado positivo, o e-mail é associado com uma certa pontuação, dependendo do teste, e vários cabeçalhos (customizáveis) são adicionados ao e-mail. A pontuação total resultante de todos os testes pode ser usada para marcar a mensagem, mover o e-mail para uma pasta específica, ou simplesmente deletar a mensagem. Alguns dos testes que o SpamAssassin realiza são os seguintes:

- Análise de cabeçalho: os spammers usam várias técnicas para mascarar suas identidades e esconder o servidor de origem de suas mensagens. O SpamAssassin tenta identificar indícios do uso desses truques;

- Análise de texto: o spam tem um estilo de texto próprio, geralmente destinado a lhe convencer de que o produto/serviço anunciado é uma oportunidade única na vida que você não pode desperdiçar, além de tentar lhe convencer de que você está recebendo esta mensagem porque se cadastrou em algum serviço ou porque um "amigo" o indicou. O SpamAssassin tenta identificar tal estilo, baseado em ocorrências comuns de palavras, frases, texto EM MAIÚSCULAS ou E N T R E S P A Ç A D O, entre outros.

- Listas negras: o SpamAssassin suporta consultas à listas negras como mail-abuse.org e ordb.org, e pode ignorar mensagens vindas de domínios reconhecidamente abusados por spammers.

- Razor: o *Vipul's Razor* é uma base de dados colaborativa para rastreamento de spam. Ela permite que um usuário reporte uma mensagem como spam, adicionando-a à base de dados do projeto, o que fará com que, automaticamente, todos os outros usuários do Razor passem a ignorar a mensagem.

No total, há mais de uma centena de testes que são executados. Uma lista completa pode ser vista em [APA 2004].

Cada teste possui um nome e uma descrição. O nome é por padrão um identificador todo em letras maiúsculas separado com o caracter "sublinha" (_), como por exemplo: "*LIMITED_TIME_ONLY*". A cada teste é associado um valor para ser incluído na pontuação, e alguns possuem um índice muito alto, como data incorreta no e-mail, domínio de DNS não existente, etc. Quando a pontuação total dos testes ultrapassa o parâmetro "*required_hits*", explícito na configuração do SpamAssassin, o e-mail é tratado como spam e reescrito de acordo com as opções. Na configuração padrão, o conteúdo do e-mail é anexado à mensagem, com um breve resumo no corpo do e-mail, e a descrição dos testes com seus resultados para classificação da mensagem. Se a pontuação final for menor do que a definida na configuração, a pontuação é passada no cabeçalho da mensagem e pode ser usada para um pós-processamento, como classificá-la como suspeita.

No cabeçalho da mensagem pode-se ver o campo "X-Spam-Status", indicando a mensagem como spam e a sua pontuação total nos testes, e no seu início, um resumo dos testes executados, e os resultados obtidos. O assunto (Subject) da mensagem também é modificado para conter o texto "*****Spam*****", o que facilita a filtragem em seu programa de e-mail.

2.3 CRM 114

FIGURA 2.3 – Logotipo CRM114

CRM114 é um sistema para examinar e-mails através de expressões regulares desenvolvido por William S. Yerazunis, e licenciado sobre a licença GPL. De acordo com o autor, CRM114 alcançou a marca de 99,87% de acertos na classificação de e-mails como spam ou não spam, conforme seus próprios testes. Nesses mesmos testes, o autor classificou manualmente essas amostras de mensagens (que eram 1518 spam e 856 não-spam, obtidos através de conjuntos de mensagens pré-classificadas na internet), e obteve 3 erros em sua classificação. Ou seja, o autor quis provar que o software que ele desenvolveu é melhor que o criador [YER 2002].

O nome CRM114 foi pego do filme "*Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb*", de Stanley Kubrick (O nome do filme no Brasil é Dr. Fantástico). No filme, a bomba chamada "*CRM114 Discriminator*" só podia ser desarmada através do recebimento de uma mensagem com o código correto de 3 dígitos. O autor aproveitou essa sigla para chamar o software de "*Controllable Regex Mutilator*".

Enquanto outros sistemas de filtros de e-mail bayesianos calculam a frequência que uma simples palavra ocorre no e-mail, CRM114 arquiva as ocorrências de frases maiores que 5 palavras em seu tamanho. Essa técnica é chamada de *Sparse Binary Polynomial Hash*, ou SPBH, uma generalização do método Baeysiano. Num nível mais baixo, o CRM114 possui uma linguagem de detecção de padrões em *strings* similar ao comando grep do unix. Por isso, ele pode ser utilizado em muitas outras aplicações, como verificação de registro de *logs* de firewalls, syslog, arquivos de dados, etc.

Exemplo em passos de como funciona o algoritimo SBPH utilizado pelo CRM114:
Passo 1: É retirado da mensagem um pedaço com N palavras;

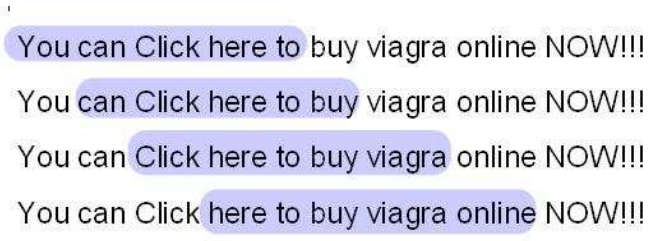

FIGURA 2.4 – Passo 1 CRM114

Passo 2: Gerado sub-frases preservando a ordem de cada pedaço de frase;

FIGURA 2.5 – Passo 2 CRM114

Passo 3: Gerado um *hash* de 32 bits das subfrases e armazenado no banco de dados, para posterior consulta.

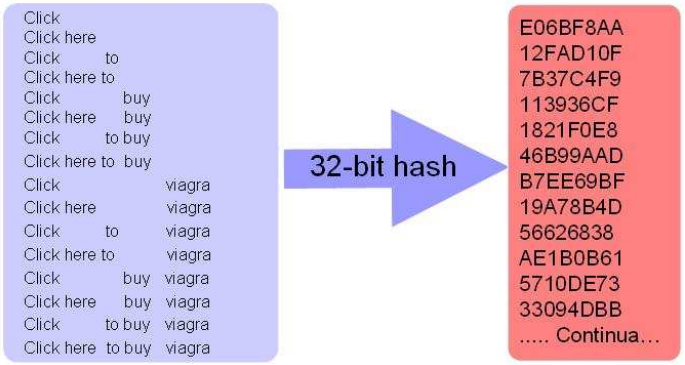

FIGURA 2.6 – Passo 3 CRM114

Dessa generalização do método bayesiano, são ainda aplicadas fórmulas estatíscas tomando o valor do hash como base, para classificar a mensagem. Para treinamento, é usado o método TOE (*Train Only Errors*), onde somente as mensagens que são classificadas incorretamente são utilizadas no treinamento. Esse método aumenta o número de chances de acerto do CRM114.

2.4 DSPAM

FIGURA 2.7 – Logotipo DSPAM

DSPAM é filtro de e-mails adaptativo feito para combater spam usando técnicas avançadas de análises estatísticas. Foi criado por Jonathan Zdziarski sob a licença GPL, e na linguagem C, para prover o máximo de performance. Em poucas palavras, DSPAM filtra spams através do aprendizado do que é e o que não é spam. Ele faz isso aprendendo pelo comportamento de usuário de e-mail. Isso habilita ao DSPAM um acerto máximo na classificação de e-mails e filtros personalizados para cada usuário. DSPAM está ganhando muito suporte na internet nos últimos anos, e já está sendo usado em implementações em larga escala, sendo alguns sistemas possuindo mais de 350.000 usuários [ZDZ 2005].

Segundo [ZDZ 2005], durante as primeiras duas semanas do período de treinamento inicial, podem ocorrer algumas ocorrências de falso positivos. Durante esse período, pode ser habilitado listas brancas (*whitelists*) para previnir falso positivos durante o treinamento.

Seu modo de funcionamento consiste na metodologia onde cada usuário de correio poderá classificar o que é o que não é spam, para a aprendizagem da ferramenta. Essa intervenção do usuário pode ser feita através do acesso a uma interface web em CGI, ou através do encaminhamento de mensagens consideradas spam pelo correio eletrônico para uma conta de correio específica, configurada pelo administrador para encaminhar ao DSPAM realizar a análise e treinamento. Essa metodologia dá aos usuários senso de participação nos esforços para combater o recebimento de spam, reduzindo a intervenção do administrador na configuração da ferramenta, enquanto as outras ferramentas concorrentes centralizam no administrador do servidor de correio o esforço de configuração. Apesar disso, o administrador pode ainda fazer treinamento da ferramenta para a base global de usuários.

DSPAM suporta os três algoritmos de combinação mais populares (*Graham-Bayesian*, *Burton-Bayesian*, e *Fisher-Robinson's Chi-Square*). Todos esses algoritmos trabalham bem para o que eles servem, e o DSPAM tem o foco de prover os melhores dados para serem tratados por esse algoritmos, utilizando técnicas de canais de *tokens*, redes neurais e redução de ruído bayesiano.

Pelo fato de Jonathan Zdziarski, desenvolvedor do DSPAM ter trabalhado junto com Bill Yerazunis no desenvolvimento do CRM114, muitos dos algoritmos do CRM114 são também utilizados no DSPAM, mas com o consentimento de Yerazunis, que disse:

> "No, it's fine. Algorithms are algorithms, and if I didn't want other people to use them, then I wouldn't have published them, or GPLed them. Jonathan has already 'made his bones' in spam fighting; he can

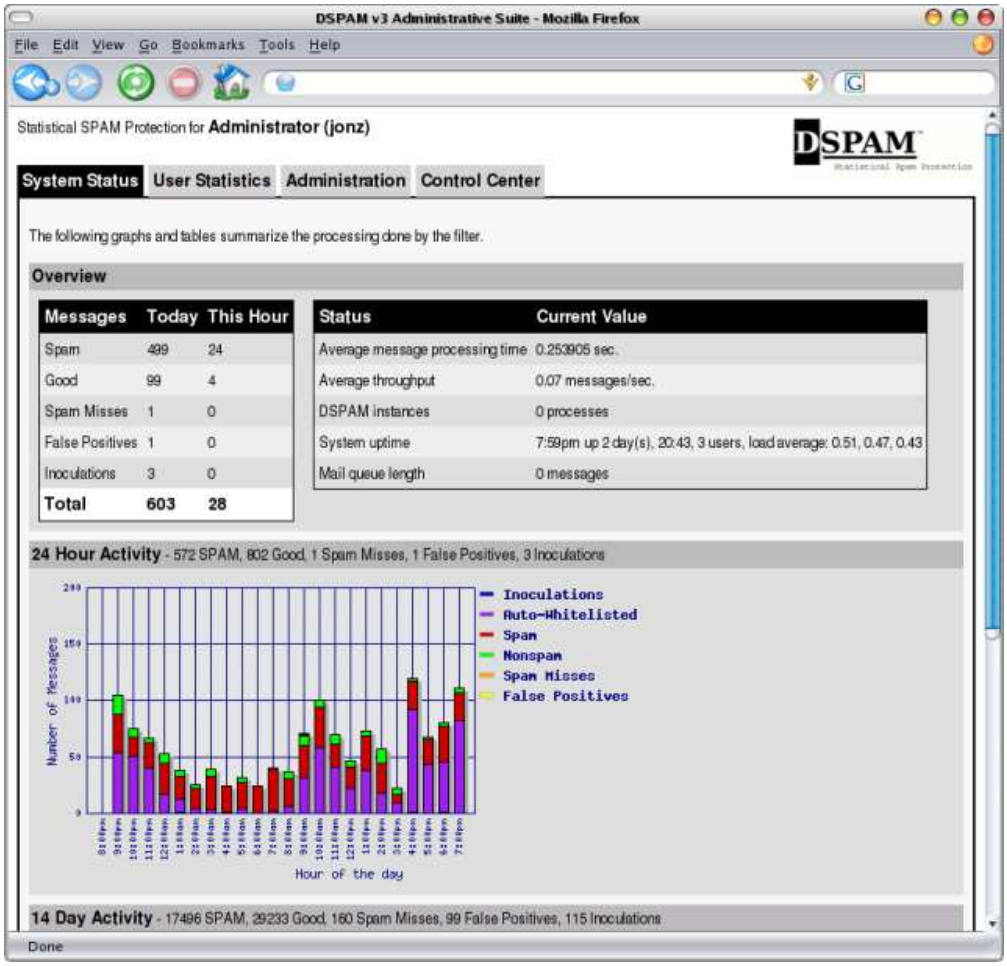

FIGURA 2.8 – Interface Web DSPAM

certainly use the theories, the algorithms, and the code. And he's even putting author credit on it. What more could you want?"

O criador do DSPAM fez questão de colocar isso no seu site em virtude de sempre ser questionado se estava roubando a pesquisa do colega Yerazunis, que como citado acima, diz que "algoritmos são algoritmos, e se ele não quisesse que fossem usados por outras pessoas, ele não os teria publicado".

2.5 Sender Policy Framework (SPF)

FIGURA 2.9 – Logotipo SPF

O SPF, ou Sender Policy Framework, foi iniciado por Meng Weng Wong, de popbox.com, em julho de 2004, com o propósito de ser um padrão da IETF para habilitar validação de fontes legítimas de e-mail. O SPF define um protocolo no qual os proprietários de domínios podem autorizar *hosts* que usem o nome do domínio na saudação smtp "MAIL FROM" ou "HELO", quando um servidor de e-mail deseja realizar uma entrega ao servidor destinatário. A decisão da política local de controle de spam se baseia no domínio do remetente da mensagem, e isso é uma vantagem porque a reputação do nome de domínio é mais confiável que a reputação do endereço IP do host. SPF torna mais difícil para spammers enviar spam, porque eles simplesmente forjavam o endereço de e-mail remetente, os quais vão ser barrados por servidores que usam SPF.

Existem várias implementações do protocolo SPF, algumas com particularidades. Será explicado nesse documento como funciona o chamado "SPF clássico", ou versão 1.

A internet usa DNS para resolver nomes de domínios (ex.:ucs.br) para endereços IP. O DNS também é utilizado para o serviço de e-mail, da seguinte forma: para cada domínio existente no mundo, um registro MX (*Mail Exchanger*) deve existir na configuração do servidor DNS de cada domínio. Um registro MX informa ao remetente do e-mail onde fica o servidor de e-mail responsável pelo recebimento de cada domínio. O SPF necessita da publicação de registros DNS chamados de "MX reversos", onde consta quais são os servidores autorizados a enviar e-mails de seu domínio. Essa entrada DNS é uma entrada TXT simples no banco de dados de cada domínio, o qual existe em todas as implantações de servidores de DNS, já que esse tipo de registro sempre fez parte do padrão do DNS [MOC 87].

Ex.:

```
; Fragmento do arquivo da zona msexchange.org
IN MX 10 mail.msexchange.org.
.....
mail IN A 192.168.1.2
; Entradas SPF
; domínio SPF
msexchange.org. IN TXT "v=spf1 mx -all"
; mail host SPF
mail IN TXT "v=spf1 a -all"
```

O servidor que recebe a mensagem realiza a verificação da autorização de envio da mensagem pelo servidor remetente durante a transação SMTP, através de três valores:

- IP: o endereço IP do cliente SMTP que está enviando o e-mail;
- Domínio: a porção do domínio da identidade "MAIL FROM" ou "HELO";
- Remetente: a identidade "MAIL FROM" ou "HELO".

Essa verificação dá um dos possíveis sete resultados:

- None: Um resultado *none* significa que não existe nenhum registro SPF registrado para o domínio. A verificação não pode afirmar se o host que está realizando a entrega está autorizado ou não.
- *Neutral*: o dono do domínio explicitou que não sabe se o IP é autorizado ou não. Um resultado neutro é tratado como o *none*.
- *Pass*: significa que o cliente é autorizado a enviar e-mail para aquele domínio contido no envelope do e-mail. As verificações de reputação podem ser realizadas, e entregar o e-mail ou não, conforme a política local (*whitelists* e *blacklists*).
- *Fail*: é um estado onde o cliente não está autorizado a enviar e-mails do domínio contido no envelope do e-mail. O software de verificação pode marcar o e-mail baseado nisso, ou simplesmente rejeitar o e-mail. Se o software decidir rejeitar o e-mail durante a transação SMTP, ele deverá retornar um código SMTP 550 [KLE 2001], e uma mensagem apropriada, como por exemplo: "*550-5.7.1 SPF MAIL FROM check failed*"
- *Softfail*: um resultado *softfail* deverá ser tratado como algo entre *fail* e *neutral*. O servidor acredita que o cliente não está autorizado, mas não segue o estado a risca. A mensagem não será rejeitada, mas será sujeita a mais alguma verificação, como por exemplo a técnica *greylisting*, onde a mensagem não será recebida na primeira vez que ela tente ser entregue, mas sim na segunda vez.
- *TempError*: significa que o cliente SPF encontrou algum problema enquanto estava realizando a verificação. O software pode escolher entre aceitar ou temporariamente rejeitar a mensagem. Se a mensagem for rejeitada durante a transação SMTP por esse motivo, o servidor deverá retornar o código SMTP 451 [KLE 2001].
- *PermError*: significa que os registros DNS publicados para o domínio não puderam ser interpretados corretamente. O servidor deve rejeitar a mensagem durante o tempo da transação SMTP, retornando o código SMTP 550 [KLE 2001].

Portanto, a implementação do SPF tem duas partes: domínios devem identificar as máquinas autorizadas a enviar e-mail em seus registros DNS, e os servidores que recebem a mensagem devem requisitar e usar a informação SPF, através de consultas DNS, que normalmente são armazenadas em cache por questões de performance, e determinar as ações dos e-mails recebidos conforme a tabela de estados listada acima.

2.6 DomainKeys

DomainKeys é um sistema anti-spam proposto pelo Yahoo! [YAH 2004] para verificação do domínio DNS de um remetente de e-mail e a integridade da mensagem. DomainKeys realiza uma função similar ao SPF em termos de prevenção de mensagens forjadas, o que não previne o spam, porém possibilita que a origem mensagem seja rastreada mais facilmente e comprovada.

O projeto Domainskeys foi iniciado no final de 2004 pelo Yahoo!. Uma vez que o Yahoo! é uma grande provedor de contas de e-mail, isso é um incentivo suficiente para desenvolvedores de software iniciarem o suporte à tecnologia em seus softwares. A licença do DomainKeys é de propriedade do Yahoo!, porém ela é compatível com a licença GPL e código aberto.

Para o dono do domínio que usa DomainKeys, existem duas vantagens:

- Permite uma redução grande no trabalho de administradores de correio que tratam o abuso de seus domínios, se os receptores do e-mail usarem o sistema de DomainKeys para eliminar automaticamente os e-mais forjados que reivindicam ser desses domínios;

- O proprietário do domínio pode então focalizar suas energias da equipe de abuso em seus próprios usuários que realmente estão abusando do uso desse domínio.

Ao mesmo tempo, existem incentivos para outros servidores de e-mail habilitarem a verificação do DomainKey:

- DomainKeys permite que a origem do e-mail seja positivamente identificado, possibilitando que o uso de listas negras e listas brancas baseados em domínio sejam mais efetivos;

- Possibilita que ataques de *phishing* sejam mais facilmente detectados;

- E-mails forjados podem ser eliminados no M.U.A. ou no M.T.A.

O protocolo DomainKeys funciona realizando um *secure hash* do conteúdo da mensagem (usando o algoritmo SHA-1 por padrão), encriptando o resultado usando uma chave privada (com o algoritimo RSA por padrão) e codificando os dados encriptados com Base64. O resultado dessa operação é adicionado ao e-mail no primeiro campo do cabeçalho SMTP com a chave “DomainKey-Signature”. Na sua essência, o processo adiciona uma assinatura digital ao e-mail.

O servidor SMTP que recebe a mensagem usa o domínio do rementente da mensagem para realizar uma pesquisa DNS na entrada TXT do domínio. O dado retornado dessa consulta de DNS é a chave pública, a qual o servidor poderá desencriptar o valor do *hash* do cabeçalho da mensagem e ao mesmo tempo recalcular o valor do *hash* do corpo da mensagem que foi recebida, a partir do ponto imediatamente seguinte ao cabeçalho ”DomainKey-Signature:”. Se os dois valores forem iguais, isso prova a um grau muito elevado de acerto de que a mensagem provém do domínio remetente, e o conteúdo da mensagem não foi alterado em trânsito.

Devido ao fato de usar cabeçalhos opcionais SMTP e entradas opcionais TXT no DNS, DomainKeys é compatível com implementações antigas de e-mail. Para Domainskeys ser utilizado na filtragem de spam, os e-mails podem ser classificados

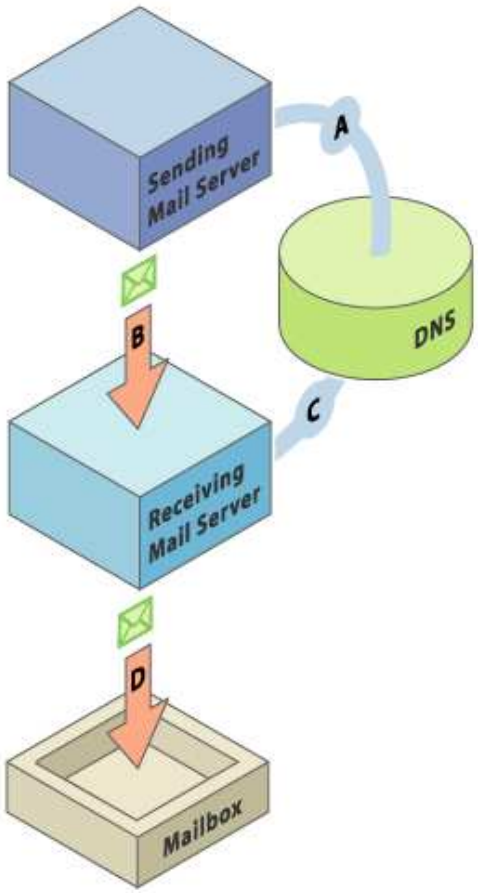

FIGURA 2.10 – Estrutura de Funcionamento do DomainKeys

em três situações, que podem serem usadas como entrada para outros algoritmos de filtragem de spam:

- Assinatura Domainkey válida: autêntico;
- Assinatura Domainkey inválida: forjado;
- Sem entrada DNS com a chave pública ou sem o cabeçalho com a entrada Domainkey: estado desconhecido.

Um dos problemas do DomainKeys é se a mensagem é modificada por um mecanismo de encaminhamento como um servidor de listas, onde isso causa a invalidez da assinatura digital no cabeçalho SMTP e a mensagem pode ser rejeitada no servidor destino. O Yahoo! sugere que as listas de e-mails reassinem e reescrevam a mensagem, ou usar o DomainKeys em combinação com outras técnicas de autenticação de remetentes, como SPF ou Sender ID (as quais também têm seus próprios problemas para lidar com o encaminhamento de e-mails).

Outro problema do DomainKeys é que cada mensagem requer um tratamento criptográfico que demanda muito uso de hardware não necessário para a entrega do e-mail. Isso é um problema sério que pode causar a não adoção desse sistema, pelo menos para os padrões de hardware atuais.

2.7 Greylisting

FIGURA 2.11 – Logotipo Greylisting

Greylisting é um método de bloquear uma parcela significante de spams ao nível do servidor de e-mail, mas sem uso de pesadas análises estatísticas ou heurísticas na mensagem. Consequentemente, suas implementações são leves, e proporcionam um decréscimo do tráfego no link e baixa carga de processamento no servidor de e-mail. O greylisting funciona através do fato de que a maioria das origens dos spams não se comporta da mesma maneira do que os servidores de e-mail considerados normais. Além de ter um grande grau de acerto no que é considerado spam, ele funciona melhor quando usado em conjunto com outros métodos de prevenção ao spam [HAR 2005A]. Seu nome deriva das palavras *whitelisting* e *blacklisting*, que significam "lista branca" e "lista negra", e isso está relacionado diretamente de como o greylisting trabalha. Como os atrasos de entrega SMTP, greylisting é um mecanismo simples mas efetivo para evitar spams através das tentativas de entrega de e-mail. A idéia é estabelecer uma relação entre o remetente e o receptor da mensagem. Se essa relação não existir previamente, a entrega da mensagem é temporariamente rejeitada com uma resposta "451 SMTP response" [KLE 2001]. Os servidores de e-mail legítimos deverão tratar essa resposta corretamente, e tentarão entregar novamente depois de algum tempo. Para estabelecer essa relação anterior, na primeira tentativa de entrega da mensagem são armazenadas a seguinte tripla:

- O remetente do e-mail;
- O endereço IP do host remetente;
- O destinatário do e-mail.

Se a entrega foi temporariamente rejeitada, essa tripla é armazenada como *greylisted*, durante um certo tempo configurável (normalmente uma hora), e depois disso essa tripla é colocada na "lista branca" (*whitelisted*). A partir desse momento, se ocorrer uma nova tentativa de entrega da mensagem, será efetivada com sucesso. Se não houver uma nova tentativa de entrega da mensagem durante um certo tempo configurável (normalmente quatro horas), a tripla expirará e será retirada do cache. Se o e-mail destino possui mais de um endereço MX para entrega (mais de um servidor de correio), a base de dados de cache do greylisting deverá ser compartilhada entre todos os servidores, para evitar que tentativas de entrega de mensagem em servidores diferentes sejam colocadas sempre no estado *greylisted*. Esse requisito de base compartilhada do cache de greylisting para mais de um servidor MX pode ocasionar um ponto de falha crítico, onde justamente mais de um MX é configurado em busca da disponibilidade do servidor de e-mail. Nesse caso, o mais recomendável é utilizar técnicas de replicação de banco de dados.

O uso do greylisting se comprovou eficiente em muitos casos, porém junto com sua aplicação, surgiram os seguintes problemas:

- Atraso na entrega de e-mails: E-mails legítimos consequentemente também sofrem atraso na sua entrega, pois dependem do servidor remetente tentar novamente realizar a entrega, e esse atraso varia de servidor a servidor. Isso pode ser evitado através do uso de *whitelists* que contém a listagem dos servidores de e-mail confiáveis e que não precisam passar pela verificação do greylisting. Uma lista usual de *whitelisting* pode ser encontrada em [HAR 2005B]. Uma maneira mais eficiente desse atraso ser evitado é integrar a técnica anti-spam SPF junto com o greylist, somente os servidores que foram identificados através da consulta SPF (conforme descrito nesse mesmo trabalho) como não sendo remetentes autorizados de envio de mensagens do domínio origem do e-mail seriam colocados em greylisting. Os servidores que foram identificados como autorizados a envio da mensagem do domínio remetente através da consulta SPF seriam considerados como *whitelist*, e não passariam pela verificação do greylist.

- Servidores diferentes de e-mail tentando enviar a mesma mensagem: Alguns domínios possuem muitos servidores que têm a tarefa de realizar a entrega da mensagem, o que pode gerar muitas triplas greylisting diferentes (o endereço IP do servidor remetente difere do já armazenado em cache), ocasionando muitas tentativas de entrega ou até mesmo o descarte da mensagem legítima. Isso pode ser evitado através de *whitelisting* como no item anterior (usando SPF ou não) ou através da alteração do uso normal do greylisting, que na tripla ao invés de armazenar somente o endereço IP do host que tentou realizar a entrega da mensagem, armazena em cache um range de ips (utilizando máscara de rede) pertencentes ao servidor remetente.

- Adaptação dos spammers ao greylisting: Sim, os remetentes de spams podem se habilitar ao greylisting e configurar seus servidores de e-mail para o reenvio da mensagem depois da primeira tentativa. Mas isso não torna o greylisting inútil. Esse atraso pode ser muito útil para que outras técnicas de spam (como spamassassin ou bogofilter, por exemplo) tenham mais chance de acerto na identificação do que é ou não é spam, utilizando um treinamento mais eficiente.

2.8 Tarpit Delay

Consiste em atrasar a entrega de mensagens de acordo com algumas políticas definidas pelo administrador do correio:

- Limite de conexões simultâneas por endereço IP: Limita o número de conexões que podem trafegar simultaneamente de um servidor origem, identificado pelo seu endereço IP. Através disso, evita-se que um servidor que esteja tentando enviar spam em massa envie todas as mensagens de uma vez só, causando atraso na entrega das demais mensagens.

- Limite de conexões com atraso: Além do limite de conexões por endereço do host de origem, após a entrega de um número X de mensagens, aumenta o tempo de espera (*delay*) de cada comando smtp das próximas mensagens vindas desse servidor origem. Usado em combinação com a validação do e-mail do destinatário,

2.9 Verificações em Transações SMTP e Cabeçalho do E-mail

Uma transação SMTP acontece quando o servidor de e-mail remetente se conecta ao servidor de e-mail destino, para realizar a entrega da mensagem. Uma vez que esse diálogo entre os servidores está em andamento, é possível realizar várias verificações, a fim de evitar a entrada de spam. As principais vantagens desse método são o pouco uso de recursos do servidor, e a economia do tráfego de internet, já que a mensagem não chega a ser entregue totalmente, caso essas verificações acusem algum problema. As verificações que podem ser feitas em uma transação smtp são as seguintes:

2.9.1 Verificação de *HELO/EHLO*

De acordo com a RFC 2821 [KLE 2001], o primeiro comando enviado pelo cliente que tenta realizar a entrega do e-mail deverá ser o *EHLO* (*HELO*, se *EHLO* não for suportado), seguido pelo seu FQDN (*Fully Qualified Domain Name*, ou nome de domínio totalmente qualificado, que representa seu endereço DNS). Esse comando é conhecido como *Hello greeting*. Se o cliente não possuir um FQDN, ele pode enviar o seu endereço IP entre colchetes, como por exemplo:"[192.168.0.4]". Essa forma é conhecida como *IPv4 address "literal" notation*. Essa sintaxe é facilmente verificada, e a mensagem pode ser rejeitada, caso não siga essas regras. Deve-se tomar cuidado para a mensagem ser rejeitada caso o cliente obedeça as regras, porém envia no *Hello greeting* o FQDN ou o endereço IP do servidor de e-mail destino.

Opcionalmente, também pode-se rejeitar clientes que não possuem o seu FQDN e tentam enviar o seu endereço IP entre colchetes, obrigando que todos os clientes que desejam enviar e-mail ao domínio destino configurado com essa opção de possuir uma entrada DNS registrada. Nesse caso, apenas seriam aceitos os *Hello greetings* com endereço IP para a rede interna. FQDNs inválidos (*hostnames* sem o domínio, ou não existentes) também devem ser rejeitados. Da mesma forma, *hostnames* que contém caracteres inválidos também devem ser rejeitados. Para domínios da internet, somente caracteres alfanuméricos e o hífen (desde que o hífem não seja o primeiro caracter) são válidos, porém alguns clientes windows também utilizam o caracter "sublinhado" (_).

É importante verificar também que a mensagem só poderá ser rejeitada após o cliente enviar o comando "RCPT TO", então o que é recomendado é impor um tempo de espera para cada comando da transação SMTP e após o "RCPT TO:", rejeitar a mensagem.

2.9.2 Verificação do DNS Reverso do Cliente

É possível verificar se o cliente rementente da mensagem possui DNS reverso configurado, e rejeitar a mensagem se o cliente não possuir. Essa configuração não é prevista na RFC 2821, porém é adotada pela maioria dos grandes provedores de internet do mundo, pressupondo-se de que se o servidor de e-mail cliente tenta entregar uma mensagem, seu DNS reverso deve existir, senão é considerado spammer.

Muitos administradores de correio tiveram problemas com esse bloqueio, mas essa exigência dos grandes provedores de internet obrigaram que essa configuração seja realizada. É muito provável que na próxima atualização da RFC 2821 isso seja

incluído.

2.9.3 Verificação do endereço do remetente

Após o *HELO/EHLO*, o cliente deverá enviar o comando smtp "MAIL FROM: <endereço>", o qual também pode ser validado. O endereço deverá estar no formato "<usuário@domínio>", e o domínio deverá existir na internet. Além dessa verificação, existe a possibilidade de mais uma, chamada *Sender Callout Verification* (no MTA Postfix, essa técnica é conhecida como *Sender Address Verification* [DON 99]). Essa verificação consiste em o servidor de e-mail que está recebendo a mensagem inicia uma segunda conexão SMTP ao servidor de e-mail do domínio enviado pelo "MAIL FROM", e tenta entregar uma mensagem para esse remetente, porém somente realiza a transação SMTP até o comando "RCPT TO:", e verifica se o servidor cliente validou a entrega da mensagem para o endereço ou não, usando esse resultado para aceitar ou não a mensagem. Porém essa última verificação pode ocasionar ataques DDoS (*Distributed Denial-of-Service*), e com o uso da técnica greylisting pelo cliente, pode rejeitar uma mensagem legítima. Por esse motivo, essa verificação não é muito utilizada, além de ser barrada em grandes provedores de internet, como a AOL, por exemplo.

2.9.4 Verificação do endereço do destinatário

A primeira e obrigatória verificação que deve ser feita no endereço do destinatário se diz respeito a prevenção do "*Open Relay*", que não permite que hosts remotos enviem e-mails para endereços remotos, ao menos que o remetente esteja devidamente autenticado. Prevenir o servidor de ser um *relay* aberto é extremamente importante, se os spammers tomarem proveito desse deslize, em muito pouco tempo o servidor estará cadastrado em diversas listas negras. Isso parece uma verificação óbvia, mas pelo que é visto no site da ORDB [ORD 2005], onde mais de duzentos mil hosts estão cadastrados como "*open-relay*", existem muitos administradores de correio descuidados, ou sem experiência:

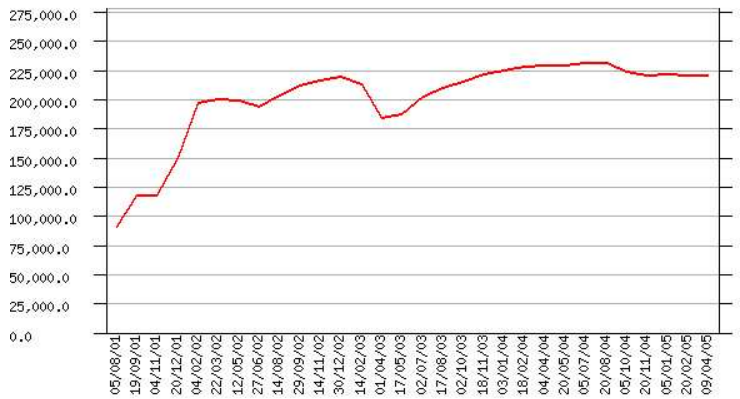

FIGURA 2.12 – Número de open-relays cadastrados na ORDB

Além da verificação do *Open Relay*, ainda é possível verificar, em tempo

da transação smtp, se o endereço do destinatário especificado no comando smtp "RCPT TO:", é um e-mail que existe no servidor onde está sendo feito a entrega da mensagem. Essa solução é simples de ser implementada em servidores que são únicos para o domínio, porém se ele for um "MX Backup" ou ele simplesmente redireciona para outro servidor de e-mail onde estão as caixas postais dos usuários, essa verificação deverá ser realizada de outra forma:

- *Recipient Callout Verification*: Na terminologia Postifix, essa verificação é chamada de *Recipient Address Verification*, em que consiste que o servidor que está recebendo a mensagem realize uma conexão SMTP com o servidor destino final e valide o destinatário nesse servidor.

- Serviços de Diretório: Uma boa solução para realizar a validação do e-mail destinatário consiste em realizar uma consulta em serviços de diretório, como o LDAP, por exemplo. A maioria dos MTA's suportam consultas a ldap.

- Se nenhuma das verificações acima for possível, pode-se gerar uma listagem das caixas postais válidas e configurar o MTA para consultar essa listagem para validar o recipiente. Essa é a solução menos elegante, pois pode ocorrer problemas na atualização dessa listagem, e demanda a realização de scripts para que o MTA seja capaz de consultar a listagem e para a geração da própria listagem.

Uma desvantagem clara da validação do endereço de destinatário é a identificação de caixas postais válidas no host destino, porém essa verificação ajuda a evitar ataques de spam baseados em dicionário, onde a lista de e-mails é gerada aleatoriamente ou através de uma listagem em poder do spammer.

2.9.5 Listas Negras

Listas negras, ou *Blacklists*, é uma base da dados dos endereços sabidos do Internet (ou dos IPs) usados pelas pessoas ou pelas companhias que emitem o spam. Há diversas organizações do Internet, possivelmente o mais proeminente os MAPS, que mantêm listas dos endereços IP que são conhecidos de alguma maneira suportar spammers (tendo relays abertos, hospedando web sites, distribuindo software de spamming marketing, etc.).

Praticamente todos os MTAs têm suporte à consulta nessas listas de uma maneira automatizada, no recibimento da mensagem. Normalmente isso é feito através de uma simples consulta de DNS baseado no endereço IP do servidor remetente da mensagem. Por exemplo, se a SPAMHAUS [SPA 2005] descobrirem que existe um servidor de correio com o *relay* aberto no endereço IP 10.20.30.40, ela irá cadastrar uma entrada DNS reversa de 40.30.20.10.sbl-xbl.spamhaus.org. Quando o servidor de correio estiver recebendo uma mensagem do endereço IP 10.20.30.40, o servidor irá consultar se existe essa entrada DNS, e se existir, indicando um spammer, irá rejeitar a mensagem. Caso a entrada DNS não existir, irá deixar a mensagem continuar a ser entregue.

As listas-negras são cadastradas de duas formas: automática, através de bancos de dados on-line mantidos por entidades sérias na Internet baseado em denúncias comprovadas; manual, através de um meio para o usuário indicar que uma mensagem é spam e, assim, seu remetente deve ser adicionado à lista-negra pessoal.

Para forçar a aceitação incondicional de determinados remetentes conhecidos, mesmo quando a origem estiver cadastrada em lista negra, as ferramentas oferecem ao usuário o recurso de cadastrar também uma lista branca (*whitelist*) com estes endereços permitidos. A lista branca tem, em geral, precedência sobre a lista negra.

Ultimamente, as listas negras públicas da internet vem sendo muito criticadas [GRA 2005], devido principalmente ao cadastro indevido de "servidores inocentes", como por exemplo ISP's que hospedam vários domínios e apenas um deles é um spammer que foi denunciado, prejudicando todos os demais clientes desses ISP's pelo fato de a consulta de blacklists ser feita através do endereço IP do servidor remetente da mensagem.

Outro problema com as listas negras é que, ao contrário de filtros, eles são controlados por humanos, o que pode ocasionar abuso de poder e influência dessas listas [GRA 2005], e essas ditas "autoridades" são pessoas desconhecidas, quem usa uma blacklist pública está confiando cegamente em pessoas que não conhece e que se acredita serem honestas, tirando o poder do administrador ou usuário decidir se irá aceitar uma mensagem ou não, e não ser bloqueada por motivos desconhecidos.

Algumas das *blacklists* públicas mais conhecidas são: ORDB, Spamcop, Spamhaus e Dsbl.

3 Implementação e Análise da Eficiência das Técnicas e Ferramentas

3.1 Descrição

Para fazer uma análise de todas as técnicas citadas, será criado um ambiente de testes onde as ferramentas para análise de conteúdo serão instaladas e configuradas conforme a necessidade. Serão feitos scripts para uso dessas ferramentas instaladas e de demais consultas que as técnicas tiverem necessidade.

Com essa análise, será possível distinguir qual das ferramentas de código aberto é mais eficaz na sua configuração padrão para evitar spam e falso-positivos através da análise de conteúdo, e qual a combinação ideal de técnicas para evitar ao máximo a utilização de recursos do servidor de correio destinatário, como uso de link internet e cpu, por exemplo. Ao mesmo tempo, evitar a ocorrência de falso-positivos, que em geral é muito mais grave do que falso-negativos, ou seja, é melhor que um spam chegue até usuário final, do que impedir que uma mensagem importante não seja lida pelo usuário porque ela foi classificada incorretamente como spam.

Essa análise também irá demonstrar se alguma técnica utilizada realmente oferece alguma vantagem de ser implementada em combinação com outra, e qual o grau de acertos e erros que uma determinada técnica proporciona na classificação de mensagens. Muitas vezes combinar algumas técnicas podem não trazer nenhum benefício a mais na filtragem de spams, apenas ocupar mais banda de *link* de internet ou tempo de processamento da mensagem.

3.2 Base de Testes

Para realizar a análise de todas as técnicas, será necessário uma base grande de e-mails pré-classificadas em spam e não-spam. Nos testes, os e-mails que não são spams serão chamados de **ham** (como utilizado na ferramenta Bogofilter), e os e-mails considerados spam serão chamados simplesmente de **spams**.

Os e-mails considerados spams foram coletados de diversas fontes públicas de spam na internet , além de uma conta de correio criada pelo autor desse documento, somente para a coleta de spams principalmente na língua portuguesa. Ao todo, serão utilizadas nos testes das técnicas **18.123** mensagens consideradas **spam**.

Para a base de e-mails consideradas não-spams (ham), foram utilizadas mensagens legitimas enviadas por colaboradores de algumas bases de listas de discussão e mensagens pessoais, além da base de mensagens do autor desse documento. Ao todo, serão utilizadas nos testes das técnicas **10.082** mensagens consideradas **não-spam**.

3.3 Ambiente de Testes

Como ambiente de testes, será utilizada a distribuição Gentoo Linux [GEN 2005], configurado e instalado em um computador Pentium 4 de 3.2 GHz com 1 Gb de memória e 120 Gb de HD. Foi escolhido o ambiente Linux pelo fato de todas as ferramentas serem facilmente instaladas e configuradas nesse ambiente, e a distribuição

Gentoo Linux para aproveitamento da sua característica de obter o máximo de performance do hardware, além de sua facilidade de instalação, personalização e atualização de pacotes pela internet.

Para realizar todos os testes, foi desenvolvido um *shell script* que realiza todas as verificações na mensagem, e guarda os resultados em um banco de dados.

Para o banco de dados onde ficará armazenado o resultado dos testes, foi utilizado o banco de dados MySQL [MYS 2005], em sua versão 4.0.24. A escolha por armazenar os resultados em banco de dados deve-se ao fato de que sua posterior consulta com diversas combinações fica facilitada, e principalmente porque a realização de todas as verificações em todas as mensagens demandam muito tempo.

3.4 Instalação das Ferramentas

As ferramentas de análise de conteúdo DSPAM, CRM114, Bogofilter e SpamAssassin foram instaladas através do gerenciador de pacotes do Gentoo Linux (*portage*), em sua configuração padrão, para que a configuração não alterasse os resultados finais dos testes.

3.5 Base de Treinamento e Configuração das Ferramentas

Como as ferramentas de análise de conteúdo Bogofilter, CRM114 e DSPAM necessitam de uma base de treinamento para criar o dicionário utilizado na classificação de e-mails, foram separados aleatoriamente da base de testes 3902 mensagens, sendo 667 consideradas não-spam e 3236 classificadas como spams. Essas mensagens separadas foram utilizadas para treinamento de cada ferramenta, de acordo com os *scripts* que se encontram em anexo à esse documento.

Para o SpamAssassin, foi utilizado um conjunto de regras para detectar também spams na língua portuguesa disponível em [LAF 2005].

O restante das configurações dessas ferramentas foram mantidas em seu modo padrão (*default*). Essa medida foi tomada para que a obtenção dos resultados refletissem a operação normal dessas ferramentas, e também não favorecer nenhuma delas.

3.6 Fase de Testes

Foi desenvolvido um *shell script* que utiliza todas as ferramentas de análise de conteúdo, e além disso, realiza os demais testes conforme a técnica anti-spam utilizada.

Devido ao fato de que os testes foram realizados em uma base estática de e-mails, e não em um servidor de correio real, as técnicas **Greylisting** e **Tarpit Delay** não foram implementadas. A técnica **Domain-Keys** também não foi implementada, em virtude de que a base de e-mails não possuía nenhuma mensagem que possuísse cabeçalho que possibilitasse o uso dessa técnica. As técnicas que utilizam análise durante a transação SMTP também não serão avaliadas, também devido ao fato de que os testes serão realizados em uma base estática de mensagens, e não em um servidor de correio real, devido a indisponibilidade de um servidor de correio

eletrônico real poder ser utilizado como ambiente de testes. Porém, conforme já descritas nesse documento, são técnicas válidas e devem ser levadas em consideração na implementação de um servidor de correio que possua técnicas de controle anti-spam.

As técnicas que foram avaliadas na fase de testes foram as seguintes:

Análise de Conteúdo:

- Dspam;
- CRM114;
- Bogofilter;
- SpamAssassin;

Análise de Cabeçalho:

- Verificação do DNS reverso;
- Verificação do domínio do rementente;
- SPF;

Listas Públicas:

- ORDB;
- Spamhaus;
- SpamCop;
- DSBL List;
- DSBL Multihop;
- DSBL Unconfirmed;

Além de verificar essas técnicas anti-spam individualmente, será verificada a combinação dessas técnicas para buscar um melhor resultado na detecção correta de spams.

As mensagens que forem classificadas pelas técnicas como spams serão chamadas simplesmente de **spams**, enquanto as mensagens que foram classificadas como mensagens legítimas (não-spam), serão chamadas de **ham**, como na ferramenta Bogofilter.

As mensagens que foram detectadas como spam pelas técnicas, porém elas foram pré-classificadas como não-spams, serão chamadas de **falso positivos**, enquanto as mensagens pré-classificadas como spam que não foram detectadas pelas técnicas como spam, serão chamadas de **falso negativos**.

3.6.1 Descrição dos Testes

Após a instalação e treinamento de todas as ferramentas de análise de conteúdo, todas as mensagens coletadas conforme descrito acima, são dispostas em dois diretórios em disco, um chamado spam e outro ham, conforme sua pré-classificação. Logo após, um *script shell* irá percorrer esses diretórios e armazenar o nome do arquivo, diretório, e uma *flag* marcando o e-mail como spam ou ham no banco de dados. Essa etapa durou em torno de uma hora para realizar a tarefa.

Após os dados de todas as mensagens coletadas serem armazenadas no banco de dados, um outro *script* realiza todos os testes em cada uma das mensagens e armazena o resultado de cada teste no banco de dados no registro que corresponde à mensagem sendo verificada. Essa etapa levou em torno de doze horas para realizar todos os testes nas 28.205 mensagens.

3.6.2 Coleta dos Resultados dos Testes

Após realizar todos os testes, o banco de dados possui todos os resultados destes testes armazenados, podendo ser facilmente verificado através de consultas em SQL, conforme anexo à esse documento. Alguns *scripts* também foram desenvolvidos visando facilitar a geração de consultas SQL.

3.7 Resultados

Ao final de todos os testes, foram coletados e tabulados todos os resultados encontrados para a avaliação das ferramentas e formulação das conclusões. Com os dados obtidos, foram geradas as seguintes tabelas e gráficos para mostrar de modo claro os resultados:

3.7.1 Resultados Individuais

TABELA 3.1 – Acertos Spam

Técnica	**Percentual de acerto**
DSPAM	99,92%
CRM114	99,55%
Bogofilter	94,28%
Spamassassin	82,65%
Verificação de DNS Reverso	30,93%
Verificação do Domínio do Remetente	7,47%
SPF	5,73%
DSBL List	5,20%
DSBL Unconfirmed	1,71%
Spamhaus	1,02%
Spamcop	0,11%
ORDB	0,04%
DSBL Multihop	0,03%

Podemos verificar na tabela 3.1 que a ferramenta **DSPAM** teve maior êxito na detecção de e-mails considerados spams, com 99,92% de detecção, seguido pelo **CRM114** com 99,55%. Isso se deve ao fato das duas técnicas compartilharem alguns algoritmos muito eficientes na detecção de spam, como o *SBPH* e *Chi-square*. Logo após veio o Bogofilter com 94,28% de acertos na detecção de spam, o que não é um resultado ruim, que pode ser aumentado com uma base maior de treinamento de seu dicionário de palavras. Por último, em ferramentas de análise de conteúdo, ficou o **SpamAssassin**, em vista de suas regras necessitarem de constante manutenção e atualização por parte do administrador do servidor de correio eletrônico.

Na verificação do cabeçalho da mensagem, a verificação do DNS reverso do endereço IP do servidor remetente da mensagem mostrou ser uma solução eficaz e com pouco uso de processamento do servidor de destino, apenas utilizando uma consulta no servidor DNS, com 30,93% de acertos de spam. Isso se deve principalmente ao fato de computadores conectados à internet serem utilizados como "zumbis", ou seja, são contaminados com vírus que tentam se alastrar enviando mensagens para vários endereços da internet aleatoriamente ou cadastrado no catálogo de endereços local da máquina. Essa técnica é muito útil para descartar mensagens consideradas spam já na transação SMTP, sem usar muitos recursos de hardware e nem do link de internet do servidor de correio que está recebendo a mensagem indesejada, além também de eliminar a mensagem antes que passe pelas ferramentas de análise de conteúdo, que

FIGURA 3.1 – Acertos Spam

demandam muito uso de hardware. Da mesma forma, a verificação da existência do domínio do remetente conseguiu barrar 7,47% das mensagens consideradas spam, apenas com uma consulta DNS. O SPF conseguiu um índice baixo de detecção de spams (5,73%), justamente pelo fato de ainda não estar implementado em muitos domínios/servidores na internet. Quando o uso dessa ferramenta for maior, e ela já é adotada por grandes provedores de internet, seu índice de acerto irá aumentar muito.

Por fim, nas consultas em listas públicas de servidores considerados spammers, a DSBL List apareceu em primeiro com um índice de acerto de 5,20%. Essa consulta em listas públicas está decaindo em seu uso, cada vez mais, pois ela já provou durante todo o tempo que essas listas existem de que não é uma forma eficiente na detecção de spams.

No gráfico 3.1, podemos verificar graficamente as técnicas e seu grau de acerto, onde a flecha simboliza o número total de spams contido na base de testes.

Podemos verificar na tabela 3.2 que um grande problema gerado pelas ferramentas de análise de conteúdo, é a ferramenta reconhecer como spam uma mensagem legítima, o conhecido falso positivo. Isso somente pode ser contornado por um grande

TABELA 3.2 – Acertos Ham

Técnica	**Percentual de acerto**
DSPAM	83,62%
CRM114	82,19%
Bogofilter	99,51%
SpamAssassin	99,65%
Verificação de DNS Reverso	97,96%
Verificação do Domínio do Remetente	99,88%
SPF	90,14%
DSBL List	94,28%
DSBL Unconfirmed	100%
Spamhaus	100%
Spamcop	100%
ORDB	100%
DSBL Multihop	100%

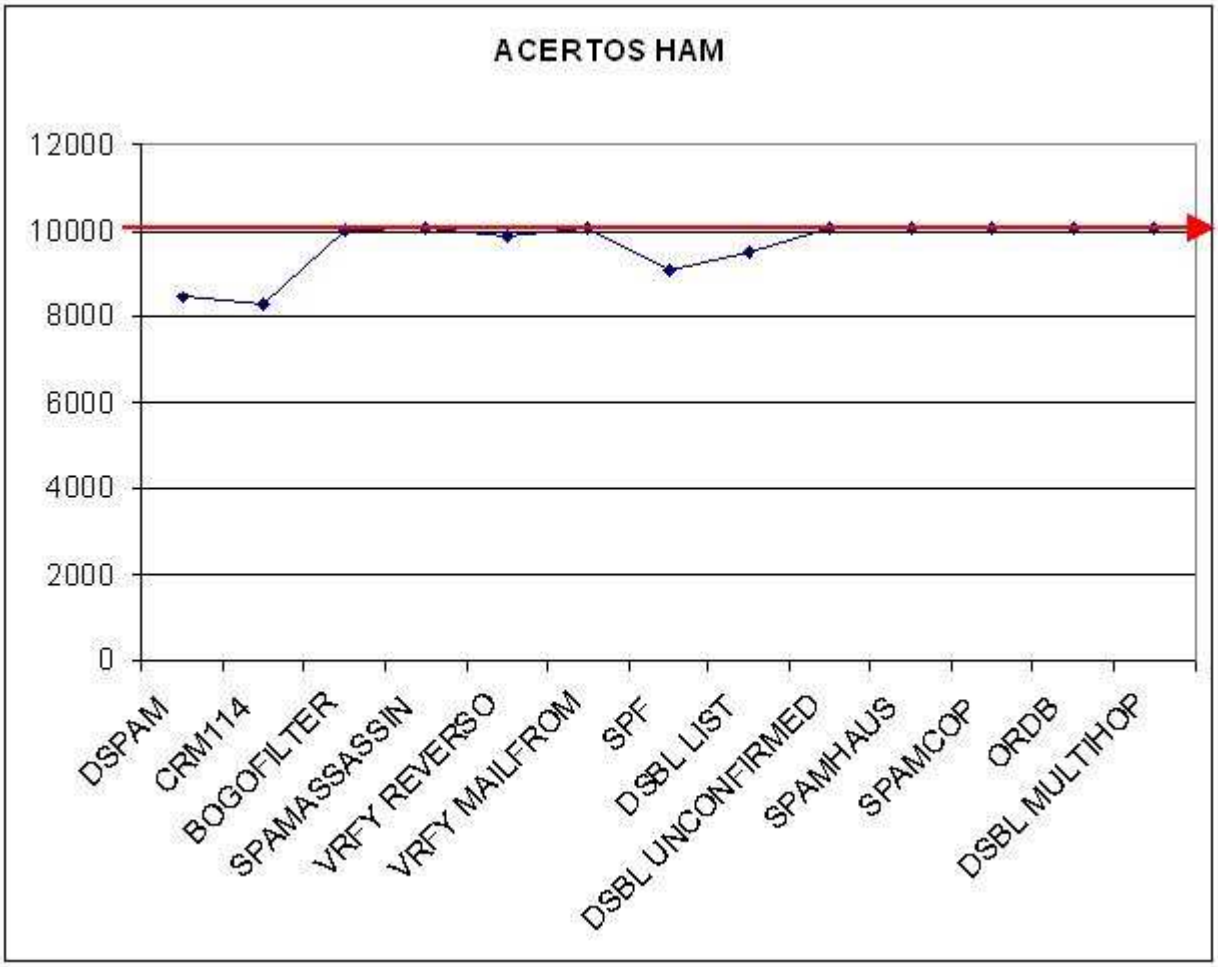

FIGURA 3.2 – Acertos Ham

e constante treinamento das ferramentas de análise das mensagens. Esse erro das ferramentas leva a um dilema: não é recomendável eliminar a mensagem classificada

como spam por essas ferramentas, e sim dar uma alternativa ao usuário ter acesso a essas mensagens, e, principalmente, reportar se a mensagem foi classificada incorretamente como spam. Isso ainda demanda muita intervenção do administrador do servidor de correio, que precisa agrupar essas mensagens erroneamente classificadas em algum local e fazer *scripts* para treinamentos das ferramentas, no caso das ferramentas CRM114 e Bogofilter. A ferramenta DSPAM tenta facilitar essa tarefa, criando uma conta de correio especial onde as mensagens classificadas incorretamente como spams são enviadas pelo usuário para posterior treinamento, porém isso causa um risco, pois leva a responsabilidade ao usuário final de classificar a mensagem, por outro lado, o usuário sente-se como colaborador na tarefa de impedir que o spam chegue até a sua caixa de mensagens. Para a ferramenta spamassassin, a única alternativa é o administrador do servidor de correio revisar suas regras ou diminuir o grau de sensibilidade para indicar se uma mensagem é spam ou não, porém essa manutenção pode provar muitos falso negativos.

TABELA 3.3 – Falso Positivos

Técnica	**Percentual**
DSPAM	5,85%
CRM114	6,37%
Bogofilter	0,17%
SpamAssassin	0,12%
Verificação de DNS Reverso	0,73%
Verificação do Domínio do Remetente	0,04%
SPF	3,52%
DSBL List	2,05%
DSBL Unconfirmed	0%
Spamhaus	0%
Spamcop	0%
ORDB	0%
DSBL Multihop	0%

Na tabela 3.3 são verificados especificamente os erros cometidos pelas técnicas, classificando como spam mensagens legítimas. A base utilizada para cálculo do percentual foi o total de mensagens, incluindo spams e não-spams. Percebe-se que as ferramentas DSPAM e CRM114 cometerem muitos erros, marcando mensagens legítimas como spam. Mas isso deve-se ao pequeno treinamento inicial com 667 mensagens que as ferramentas receberam para realizar a análise, no próprio site da ferramenta DSPAM [ZDZ 2005] orienta que nas primeiras duas semanas de uso da ferramenta podem ocorrer muitos falso-positivos, já que a ferramenta aprende de acordo com o uso de cada usuário, exigindo uma pré-classificação de 2500 mensagens para um maior grau de acerto. No caso da ferramenta CRM114, o sei treinamento se dá através dos erros (*train-on-error*), conforme já foi detalhado anteriormente. O alto grau de falso positivos dessas duas ferramentas então ocorre devido ao pouco treinamento, e não pelo fato dessas ferramentas não sejam eficientes.

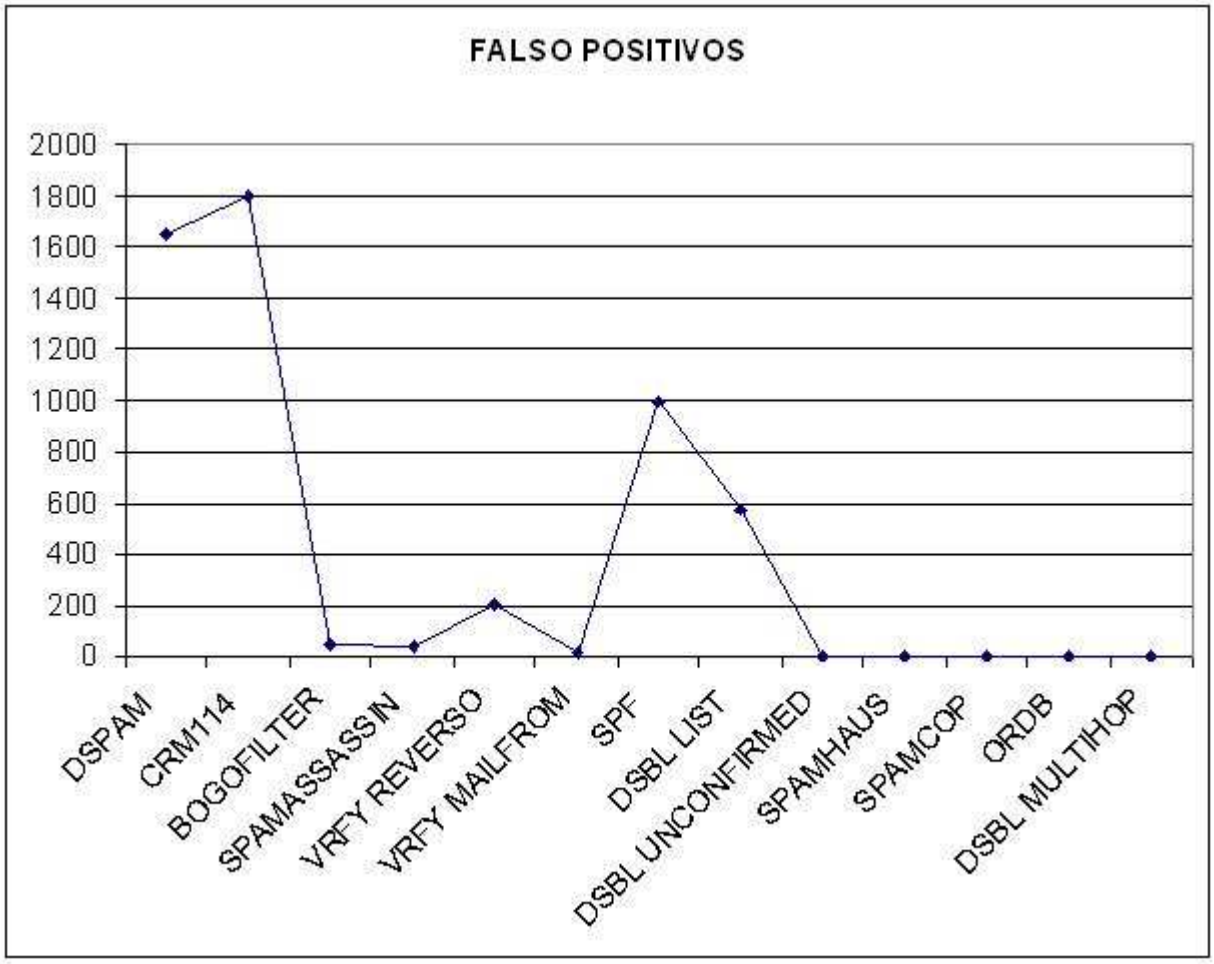

FIGURA 3.3 – Falso Positivos

TABELA 3.4 – Falso Negativos

Técnica	**Percentual**
DSPAM	0,05%
CRM114	0,29%
Bogofilter	3,68%
SpamAssassin	11,15%
Verificação de DNS Reverso	44,38%
Verificação do Domínio do Remetente	59,46%
SPF	60,57%
DSBL List	60,91%
DSBL Unconfirmed	63,16%
Spamhaus	63,60%
Spamcop	64,18%
ORDB	64,23%
DSBL Multihop	64,23%

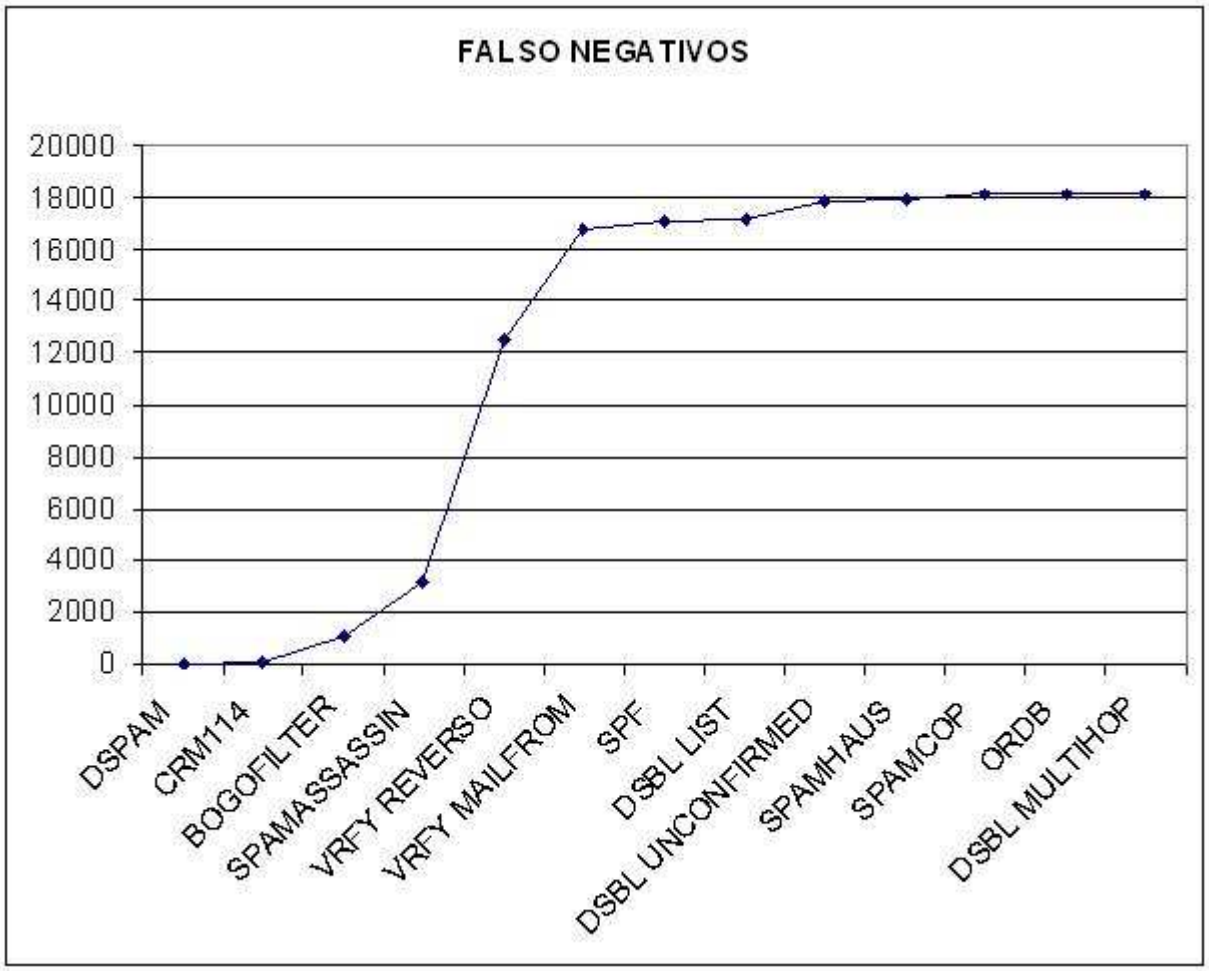

FIGURA 3.4 – Falso Negativos

Na tabela 3.4 pode-se observar especificamente os erros cometidos pelas técnicas classificando como não spam mensagens pré-classificadas como spam. A base utilizada para o cálculo do percentual foi o total de mensagens, e a ferramenta que menos errou na classificação da mensagem como spam foi a ferramenta DSPAM.

A tabela 3.5 contém o percentual de erros de classificação de cada técnica avaliada (soma dos falso positivos com falso negativos). A técnica que menos teve erros foi a Bogofilter, que obteve 3,85% de erros na sua classificação, porém como a maioria dos seus erros foi a de falso negativos, a sua utilização individual e com pouco treinamento pode levar a classificação incorreta das mensagens. As ferramentas DSPAM e CRM114 tiveram um índice alto de falso-positivos, e-mails legítimos que podem ser considerados importantes para a organização, ocasionando um grande transtorno para o destinatário da mensagem, maior até mesmo do causado pelo recebimento de spam. A não entrega de apenas um e-mail legítimo pode causar uma grande perda financeira para a organização, portanto o objetivo maior é como classificar corretamente as mensagens como spam, porém evitando ao máximo a ocorrência de falso positivos. A resposta para isto está na combinação de das técnicas, de uma maneira que uma técnica que tenha o ponto forte no reconhecimento de mensagens

TABELA 3.5 – Erros

Técnica	**Percentual**
DSPAM	5,90%
CRM114	6,65%
Bogofilter	3,85%
SpamAssassin	11,27%
Verificação de DNS Reverso	45,11%
Verificação do Domínio do Remetente	59,50%
SPF	64,10%
DSBL List	62,96%
DSBL Unconfirmed	63,16%
Spamhaus	63,60%
Spamcop	64,18%
ORDB	64,23%
DSBL Multihop	64,23%

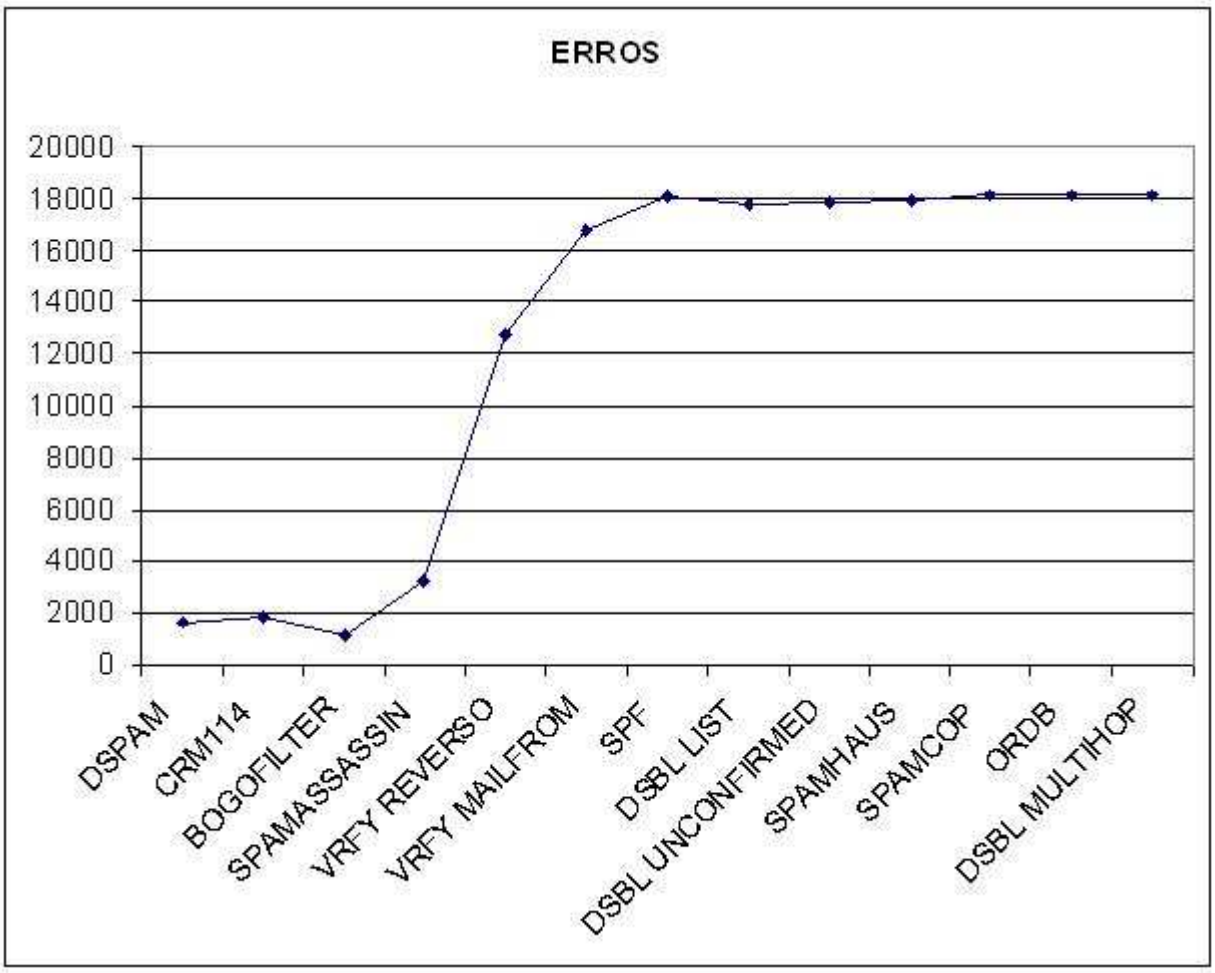

FIGURA 3.5 – Erros

legítimas (Ex.: SPF, Greylisting) crie uma pré-classificação nas mensagens que tenham grande chance de serem não-spam, evitando que as ferramentas de análise

de conteúdo a classifiquem incorretamente.

Por fim, segue o gráfico com o resultado de todas as técnicas testadas individualmente para análise.

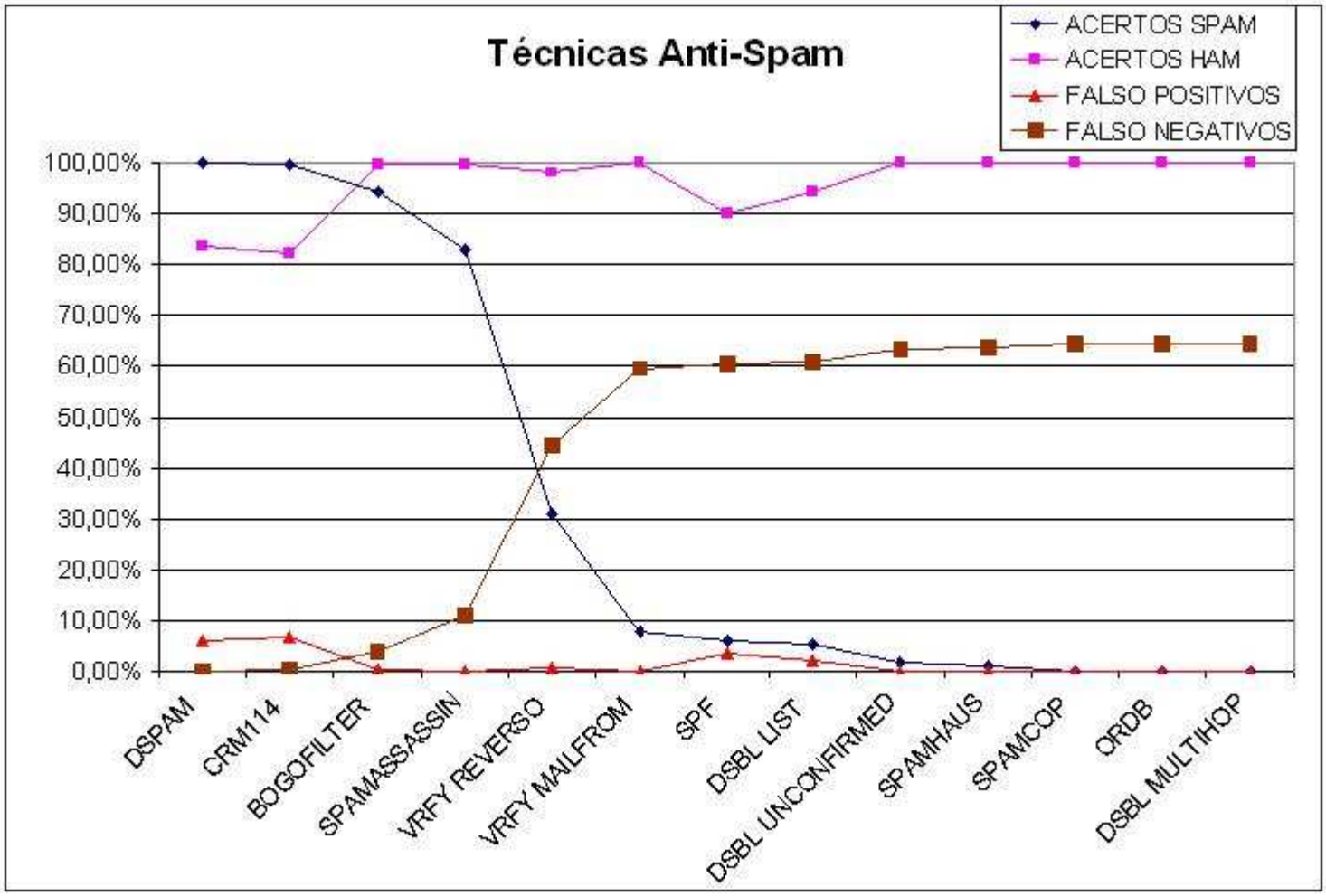

FIGURA 3.6 – Análise Individual

3.7.2 Resultados Combinando Duas Técnicas

Esses testes foram realizados combinando-se sempre uma técnica de análise de conteúdo junto com uma técnica de análise de cabeçalho ou uma técnica de consulta à listas públicas. Não foram combinadas duas técnicas de análise de conteúdo porque isso não é recomendável, pois além de gerar muita exigência do hardware, poderá gerar muitos falso positivos, o que não é aceitável. Segue na tabela os resultados dos testes usando todas as possibilidades de combinação de duas técnicas levando em consideração que duas técnicas de análise de conteúdo não serão testadas juntas.

TABELA 3.6 – Combinação de DSPAM com outra Técnica

Combinação DSPAM	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
DSPAM + Vrf. Reverso	99,95%	83,16%	6,02%	0,03%	6,05%
DSPAM + DSBL Unconfirmed	99,93%	83,62%	5,85%	0,05%	5,90%
DSPAM + DSBL List	99,93%	78,25%	7,78%	0,05%	7,82%
DSPAM + SPF	99,92%	90,14%	3,52%	0,05%	3,57%
DSPAM + DSBL Multihop	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + Spamcop	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + ORDB	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + Spamhaus	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + Vrf. Mailfrom	99,92%	83,54%	5,89%	0,05%	5,94%

Podemos verificar na tabela 3.6 que o DSPAM combinado com a técnica SPF obteve o número mais baixo de falso-positivos, que foi de 3,52%. Apesar desse valor ainda ser alto, o SPF consegue evitar que o DSPAM classifique erroneamente algumas mensagens legítimas como spam, mesmo não contribuindo para o grau de acerto de mensagens spam, porém evitando que a análise seja feita pelo DSPAM, que consome mais recursos de hardware que o SPF, inclusive tendo uma econocmia no link de internet, já que para análise de SPF seja realizada, não é necessário que toda a mensagem seja entrega ao servidor destino para ser analisada. Nas demais combinações, a análise teve resultados muito próximos entre si no acerto de classificação das mensagens como spam, porém ocorreu um percentual maior de falso positivos.

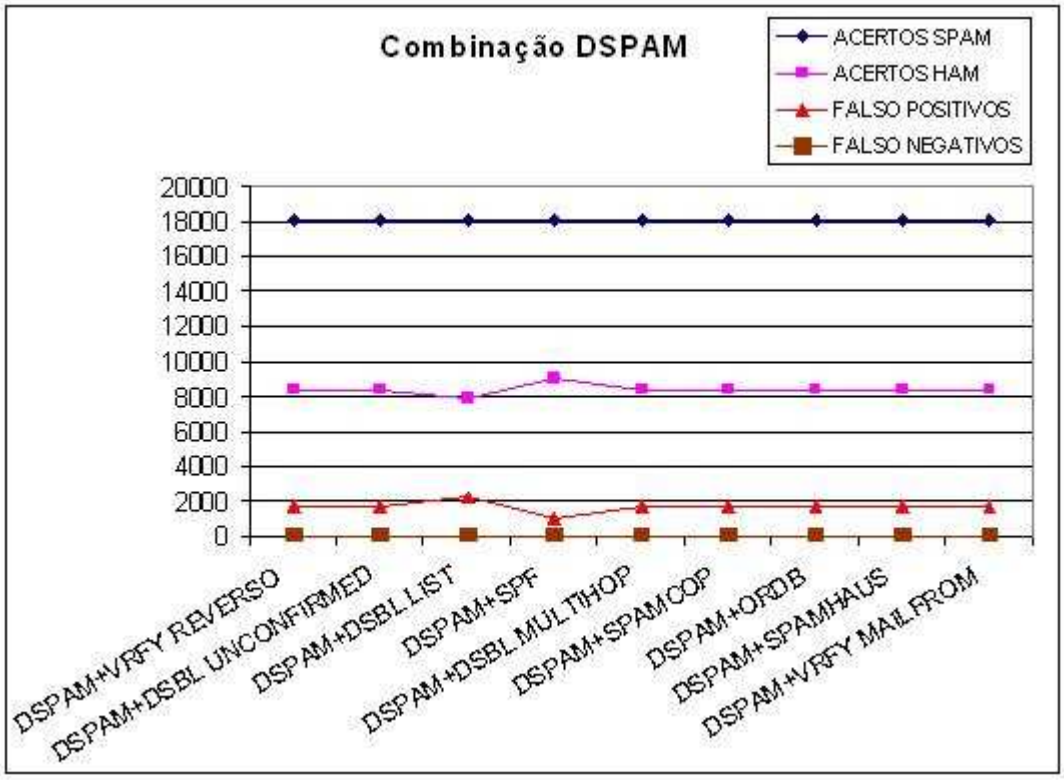

FIGURA 3.7 – Combinação de DSPAM com outra Técnica

TABELA 3.7 – Combinação de CRM114 com outra Técnica

Combinação CRM114	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
CRM114 + Verif. Reverso	99,88%	81,22%	6,71%	0,08%	6,79%
CRM114 + DSBL List	99,66%	76,72%	8,32%	0,22%	8,54%
CRM114 + DSBL Unconf.	99,56%	82,19%	6,37%	0,28%	6,65%
CRM114 + Verif. Mailfrom	99,56%	82,09%	6,40%	0,28%	6,68%
CRM114 + Spamcop	99,55%	82,19%	6,37%	0,29%	6,65%
CRM114 + DSBL Multihop	99,55%	82,19%	6,37%	0,29%	6,65%
CRM114 + ORDB	99,55%	82,19%	6,37%	0,29%	6,65%
CRM114 + Spamhaus	99,55%	82,19%	6,37%	0,29%	6,65%

Na tabela 3.7 é verificado que se o administrador optar de usar a ferramenta CRM114, será necessário uma grande intervenção (ao menos inicialmente) no treinamento nos erros que ela gera. Os índices de falso-positivos foram altos e a combinação

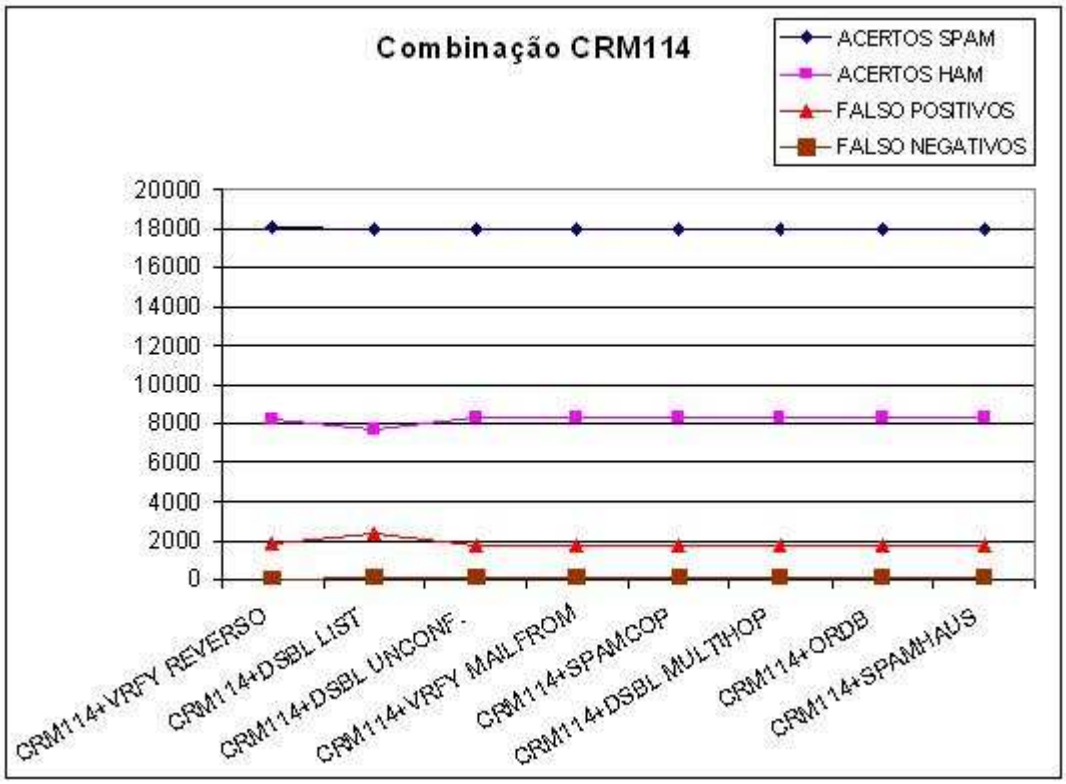

FIGURA 3.8 – Combinação de CRM114 com outra Técnica

com outras técnicas não conseguiram diminuir esses índices.

TABELA 3.8 – Combinação de Bogofilter com outra Técnica

Combinação Bogofilter	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
Bogofilter + Verif. Reverso	96,82%	97,54%	0,88%	2,05%	2,93%
Bogofilter + DSBL List	95,23%	93,79%	2,22%	3,06%	5,28%
Bogofilter + Verif. Mailfrom	94,74%	99,39%	0,22%	3,38%	3,60%
Bogofilter + DSBL Unconf.	94,71%	99,51%	0,17%	3,40%	3,57%
Bogofilter + Spamhaus	94,42%	99,51%	0,17%	3,59%	3,76%
Bogofilter + Spamcop	94,29%	99,51%	0,17%	3,67%	3,84%
Bogofilter + DSBL Multihop	94,28%	99,51%	0,17%	3,68%	3,85%
Bogofilter + ORDB	94,28%	99,51%	0,17%	3,68%	3,85%

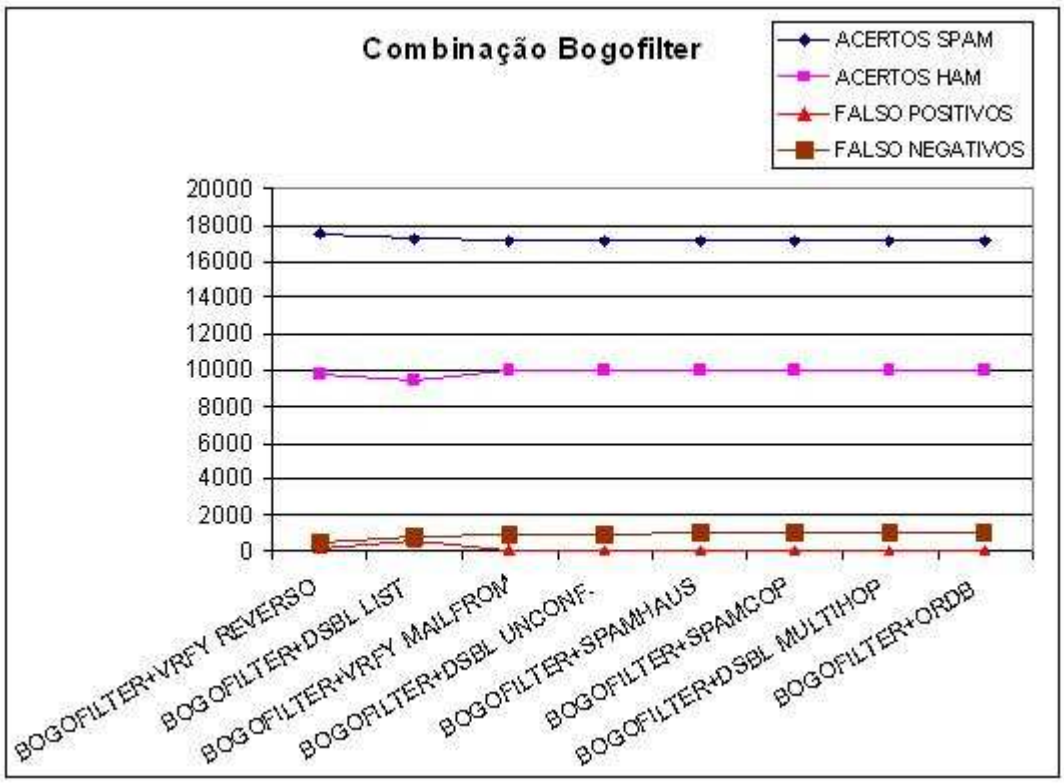

FIGURA 3.9 – Combinação de Bogofilter com outra Técnica

A combinação da ferramenta Bogofilter com outra técnica de análise do cabeçalho da mensagem teve bons resultados, chegando ao acerto de spam em 96,82% e acertos de hams em 97,54% na combinação com a verificação do DNS reverso do remetente da mensagem, como relacionado na Na tabela 3.8. São ótimos resultados para uma configuração padrão e com o pouco treinamento que a ferramenta teve para realização dos testes. Porém a detecção de spams mais complexos como as diversas combinações da palavra VIAGRA, por exemplo, fica comprometida devido o dicionário de palavras do Bogofilter ser formado por entradas de palavras simples. Seu uso irá demandar sempre intervenção pelo administrador do servidor de correio, que deverá sempre realizar o treinamento da ferramenta com novos spams que surgem a cada dia.

A melhor combinação encontrada do SpamAssassin foi com a verificação do DNS reverso, com 88,64% de acerto de mensagens spams e 97,61% não-spams, como demonstrado na tabela 3.9. É um grau de acerto mediano, porém o uso de muitos recursos de hardware por essa ferramenta e a constante intervenção por parte do administrador nas regras (que não são muito simples de configurar, é necessário um grande conhecimento de expressões regulares) dificultando a sua manutenção. Apesar de existirem vários sites compartilhando essas regras do SpamAssassin, o aumento do número de regras pode tornar inviável a sua implantação em servidores de alto tráfego de e-mail, devido ao grande uso de CPU.

TABELA 3.9 – Combinação de SpamAssassin com outra Técnica

Combinação SpamAssassin	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
SpamAssassin + Verif. Reverso	88,64%	97,61%	0,85%	7,30%	8,15%
SpamAssassin + DSBL List	84,14%	93,93%	2,17%	10,19%	12,36%
SpamAssassin + Verif. Mailfrom	83,67%	99,53%	0,17%	10,49%	10,66%
SpamAssassin + Spamhaus	82,68%	99,65%	0,12%	11,13%	11,25%
SpamAssassin + DSBL Unconf.	82,66%	99,65%	0,12%	11,14%	11,26%
SpamAssassin + ORDB	82,66%	99,65%	0,12%	11,14%	11,27%
SpamAssassin + DSBL Multihop	82,66%	99,65%	0,12%	11,14%	11,27%
SpamAssassin + Spamcop	82,66%	99,65%	0,12%	11,14%	11,27%

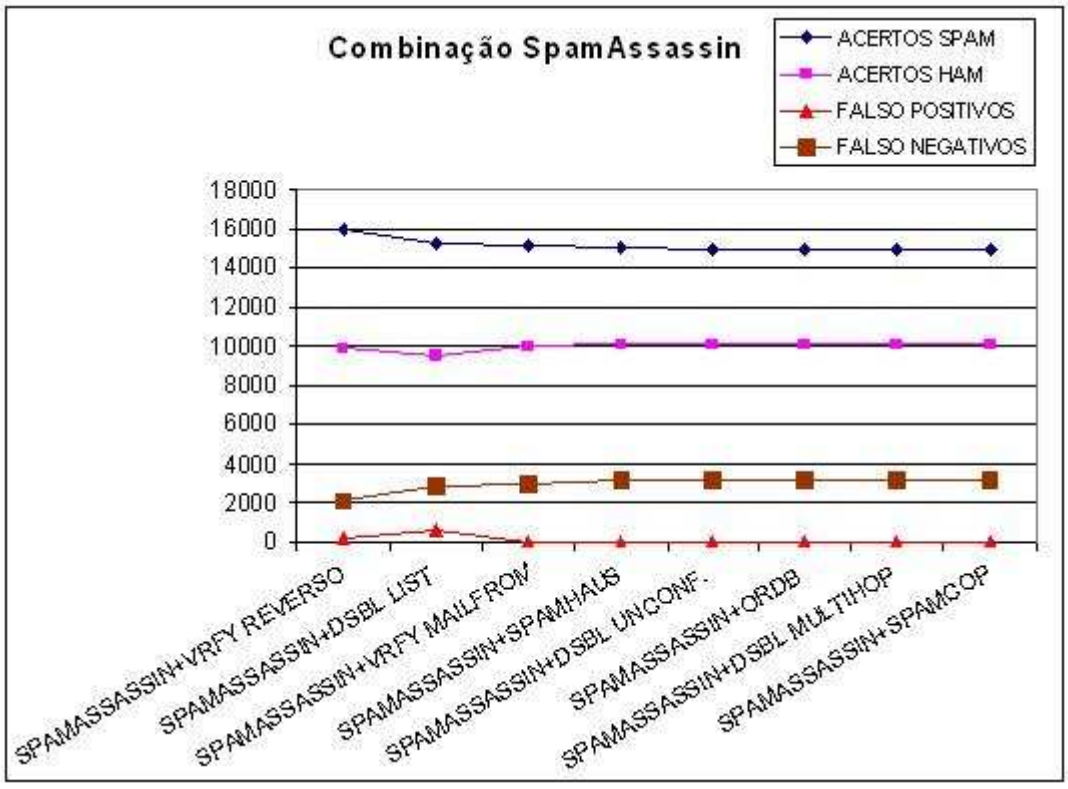

FIGURA 3.10 – Combinação de SpamAssassin com outra Técnica

Na tabela 3.10 estão os resultados combinando-se a técnica de verificação do DNS reverso com outras técnicas de análise do cabeçalho. Sendo que indi-

TABELA 3.10 – Combinação de Verificação de DNS Reverso com outra Técnica

Combinação com Verif. DNS Reverso	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
Verif. Reverso + Verif. Mailfrom	37,50%	97,84%	0,77%	40,16%	40,93%
Verif. Reverso + DSBL List	32,00%	92,24%	2,77%	43,69%	46,47%
Verif. Reverso + DSBL Unconf.	31,91%	97,96%	0,73%	43,75%	44,48%
Verif. Reverso + Spamhaus	31,53%	97,96%	0,73%	44,00%	44,73%
Verif. Reverso + Spamcop	30,98%	97,96%	0,73%	44,35%	45,08%
Verif. Reverso + DSBL Multihop	30,96%	97,96%	0,73%	44,36%	45,09%
Verif. Reverso + ORDB	30,95%	97,96%	0,73%	44,37%	45,10%

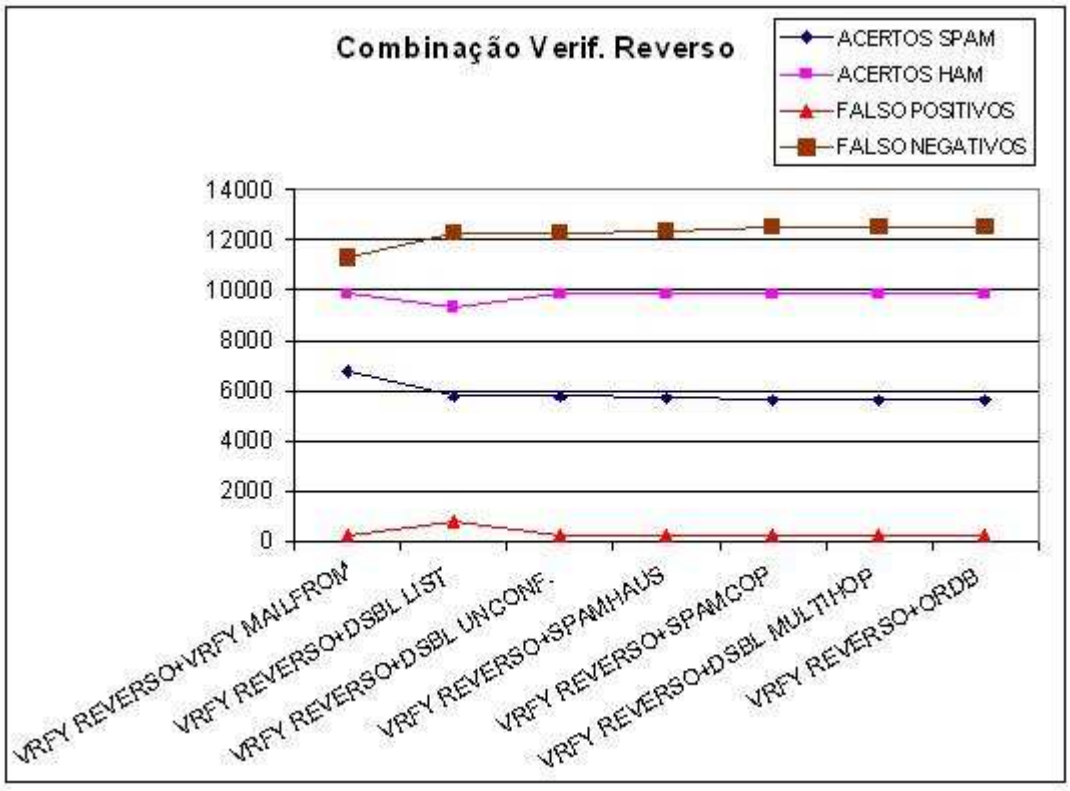

FIGURA 3.11 – Combinação de Verificação de DNS Reverso com outra Técnica

vidualmente esta técnica teve um acerto de classificação de spams de 30,93%, a combinação com a verificação se o domínio do remetente existe aumentou esse índice para 37,5%, o que é um resultado nada desprezível, em vista da simplicidade dessa verificação. O número de falso positivos de 0,77% nessa combinação deve-se

à servidores mal configurados, que não possue seu DNS reverso cadastrado como deveriam ser, conforme recomenda a RFC 2505 [LIN 99]. A utilização dessas duas técnicas combinadas diminuem bastante a chance de que um spam utilize recursos do link de internet, além de diminuir a carga de algum outro teste na análise de conteúdo que possa eventualmente ser realizada após a entrega da mensagem.

TABELA 3.11 – Combinação de Verificação de MailFrom com outra Técnica

Combinação com Verif. MailFrom	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
Verif. MailFrom + DSBL List	12,54%	94,16%	2,09%	56,20%	58,29%
Verif. MailFrom + DSBL Unconf.	9,09%	99,88%	0,04%	58,41%	58,45%
Verif. MailFrom + Spamhaus	8,46%	99,88%	0,04%	58,82%	58,86%
Verif. MailFrom + Spamcop	7,56%	99,88%	0,04%	59,39%	59,44%
Verif. MailFrom + DSBL Multihop	7,50%	99,88%	0,04%	59,44%	59,48%
Verif. MailFrom + ORDB	7,50%	99,88%	0,04%	59,44%	59,48%

Conforme os resultados na tabela 3.11, a utilização dessa técnica sozinha sem a verificação do reverso não traz vantagens já que nos spams a maioria dos e-mails remetentes são forjados.

Como listado na tabela 3.12, o máximo de acertos na classificação de spam foi de 5,91%, um índice baixo. Pelo que demonstra, a utilização de listas negras públicas está perdendo terreno para lista negras privadas junto de técnicas de confirmação da autenticidade do remetente, como SPF e DomainKeys.

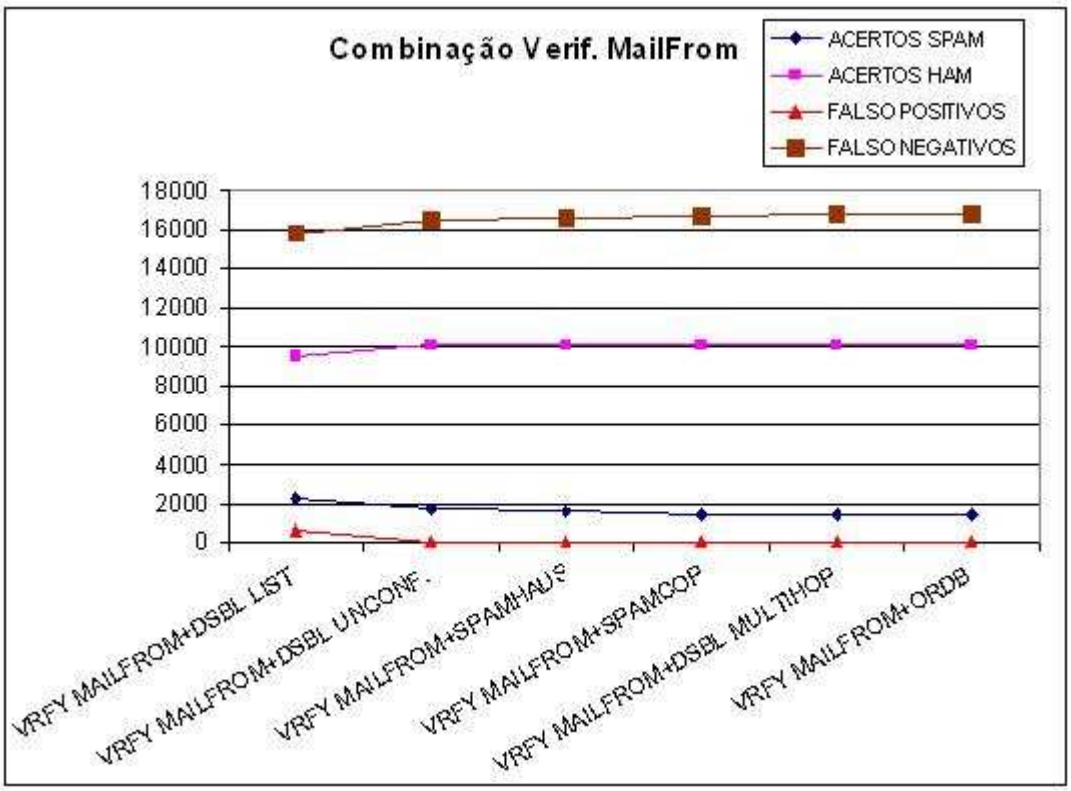

FIGURA 3.12 – Combinação de Verificação de MailFrom com outra Técnica

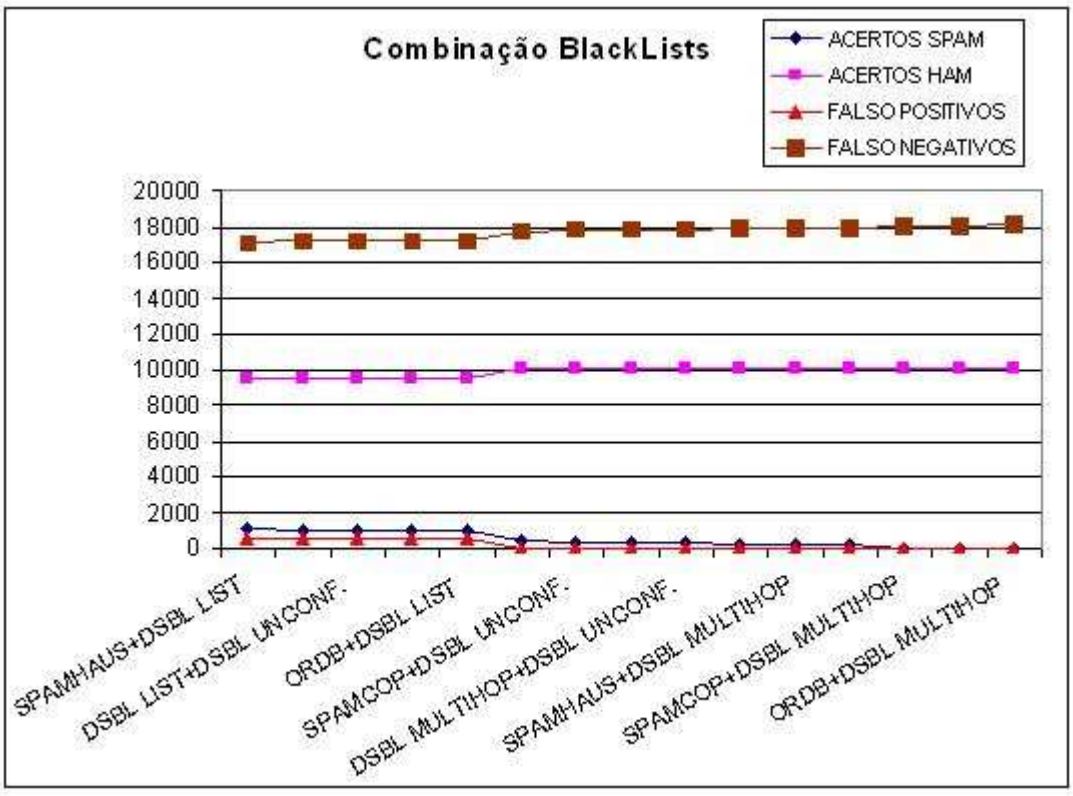

FIGURA 3.13 – Combinação de Verificação de Listas Negras com outra Técnica

TABELA 3.12 – Combinação de Verificação de Listas Negras com outra Técnica

Combinação com Listas Negras	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
Spamhaus + DSBL List	5,91%	94,28%	2,05%	60,46%	62,50%
Spamcop + DSBL List	5,28%	94,28%	2,05%	60,86%	62,91%
DSBL List + DSBL Unconf.	5,24%	94,28%	2,05%	60,89%	62,94%
DSBL List + DSBL Multihop	5,24%	94,28%	2,05%	60,89%	62,94%
ORDB + DSBL List	5,22%	94,28%	2,05%	60,90%	62,95%
Spamhaus + DSBL Unconf.	2,44%	100,00%	0,00%	62,68%	62,68%
Spamcop + DSBL Unconf.	1,79%	100,00%	0,00%	63,11%	63,11%
ORDB + DSBL Unconf.	1,73%	100,00%	0,00%	63,14%	63,14%
DSBL Multihop + DSBL Unconf.	1,72%	100,00%	0,00%	63,15%	63,15%
ORDB + Spamhaus	1,06%	100,00%	0,00%	63,57%	63,57%
Spamhaus + DSBL Multihop	1,05%	100,00%	0,00%	63,58%	63,58%
Spamcop + Spamhaus	1,05%	100,00%	0,00%	63,58%	63,58%
Spamcop + DSBL Multihop	0,14%	100,00%	0,00%	64,16%	64,16%
ORDB + Spamcop	0,14%	100,00%	0,00%	64,17%	64,17%
ORDB + DSBL Multihop	0,07%	100,00%	0,00%	64,21%	64,21%

3.7.3 Resultados Combinando Três ou Mais Técnicas

Para realizar a combinação de três ou mais técnicas, foi selecionado somente o DSPAM como técnica de análise de conteúdo, pelo fato de que essa ferramenta foi a que conseguiu melhor resultado para identificação de spam, em comparação com Bogofilter, CRM114 e SpamAssassin. Além disso, ela é a ferramenta que se mostrou mais fácil de administrar e o fato de envolver o usuário final no treinamento da ferramenta é um diferencial que a destaca entre as demais.

Porém, o número alto de falso-positivos preocupa, mas em combinação com outras técnicas isso pode ser reduzido, como veremos nos resultados que seguem:

Na tabela 3.13 se observa que a combinaçao de DSPAM, verificação do DNS reverso e SPF é a mais interessante, devido ao alto índice de acerto de spams e o nível de falso-positivos diminuiu bastante. Nota-se também que já existem domínios com entrada SPF que enviam spam, o que estimula o uso de blacklists ou futuramente listas públicas de reputação de servidores de e-mail. Conforme o número de servidores adotem as técnicas de SPF ou Domainkeys, técnicas que garantem a autenticidade da origem do e-mail, torna-se mais fácil identificar os que podem ser considerados spammers, estimulando o uso de listas negras ou o uso dos futuros servidores de reputação.

Na tabela 3.14 estão os resultados de outras combinações possíveis de técnicas, porém com o número de falso-positivos muito alto, a não ser a combinação “DSPAM + verificação de mailfrom + SPF”, que manteve o índice de falso-positivos baixo.

As demais combinação de técnicas presentes na tabela 3.15 não apresentaram uma melhora significativa no número de falso-positivos, mantendo os resultados parecidos entre as combinações.

TABELA 3.13 – Combinação DSPAM + Várias Técnicas - 1

Combinação	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
DSPAM + Verif.Reverso + DSBL Unconf.	99,96%	83,16%	6,02%	0,03%	6,05%
DSPAM + Verif.Reverso + DSBL List	99,96%	77,78%	7,94%	0,03%	7,97%
DSPAM + Verif.Reverso + Verif. Mailfrom + SPF	99,95%	90,14%	3,52%	0,03%	3,56%
DSPAM + Verif.Reverso + SPF	99,95%	90,14%	3,52%	0,03%	3,56%
DSPAM + ORDB + Verif.Reverso	99,95%	83,16%	6,02%	0,03%	6,05%
DSPAM + Verif.Reverso + DSBL Multihop	99,95%	83,16%	6,02%	0,03%	6,05%
DSPAM + Verif.Reverso + Spamcop	99,95%	83,16%	6,02%	0,03%	6,05%
DSPAM + Verif.Reverso + Spamhaus	99,95%	83,16%	6,02%	0,03%	6,05%
DSPAM + Verif.Reverso + Verif. Mailfrom	99,95%	83,07%	6,05%	0,03%	6,08%
DSPAM + ORDB + DSBL Unconf.	99,93%	83,62%	5,85%	0,05%	5,90%
DSPAM + DSBL Multihop + DSBL Unconf.	99,93%	83,62%	5,85%	0,05%	5,90%

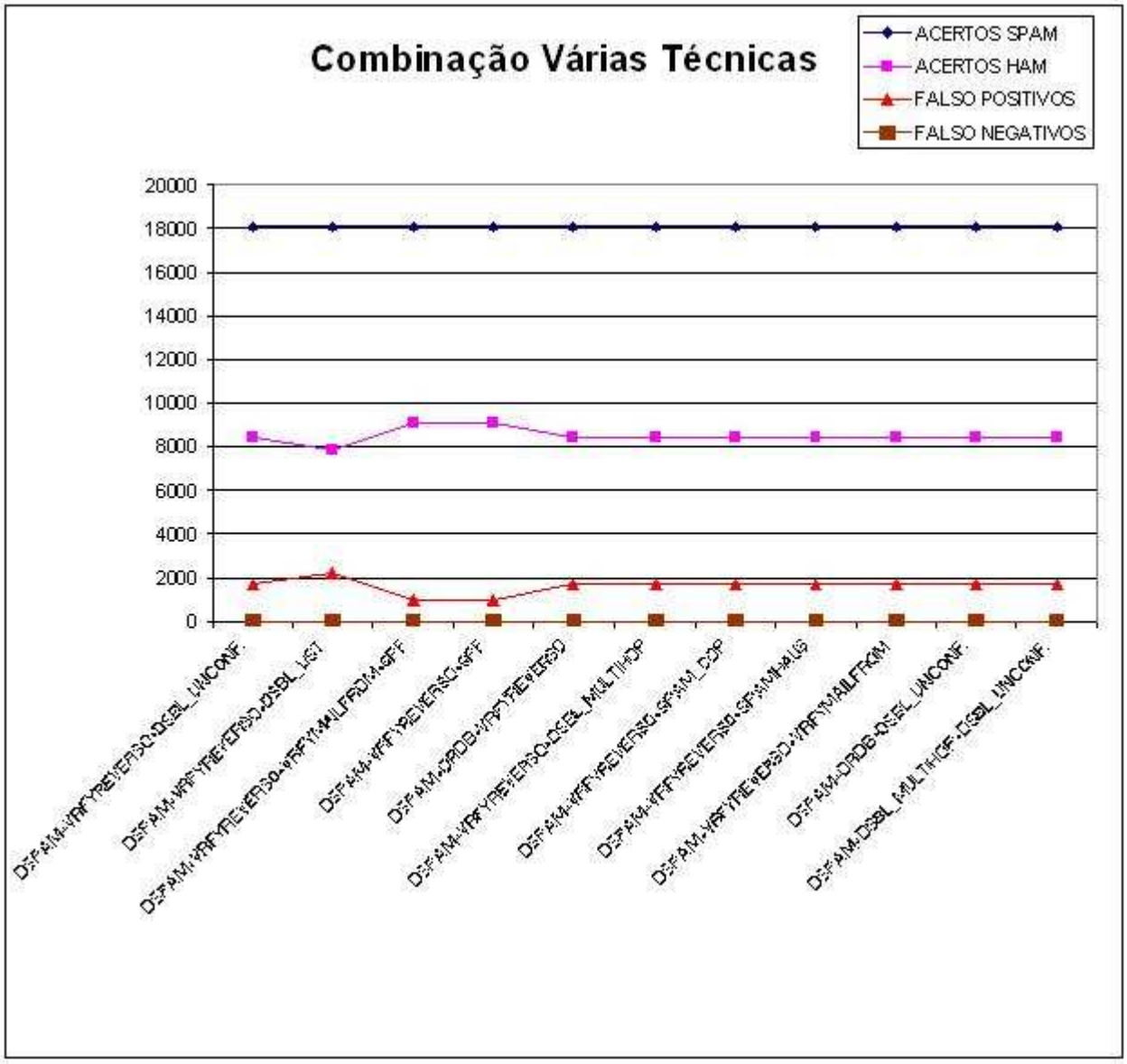

FIGURA 3.14 – Combinação DSPAM + Várias Técnicas - 1

TABELA 3.14 – Combinação DSPAM + Várias Técnicas - 2

Combinação	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
DSPAM + Spamcop + DSBL Unconf.	99,93%	83,62%	5,85%	0,05%	5,90%
DSPAM + Spamhaus + DSBL Unconf.	99,93%	83,62%	5,85%	0,05%	5,90%
DSPAM + Verif. Mailfrom + DSBL Unconf.	99,93%	83,54%	5,89%	0,05%	5,93%
DSPAM + ORDB + DSBL List	99,93%	78,25%	7,78%	0,05%	7,82%
DSPAM + DSBL List + DSBL Unconf.	99,93%	78,25%	7,78%	0,05%	7,82%
DSPAM + Spamhaus + DSBL List	99,93%	78,25%	7,78%	0,05%	7,82%
DSPAM + DSBL List + DSBL Multihop	99,93%	78,25%	7,78%	0,05%	7,82%
DSPAM + Spamcop + DSBL List	99,93%	78,25%	7,78%	0,05%	7,82%
DSPAM + Verif. Mailfrom + DSBL List	99,93%	78,16%	7,81%	0,05%	7,85%
DSPAM + Verif. Mailfrom + SPF	99,92%	90,14%	3,52%	0,05%	3,57%
DSPAM + Spamcop + DSBL Multihop	99,92%	83,62%	5,85%	0,05%	5,90%

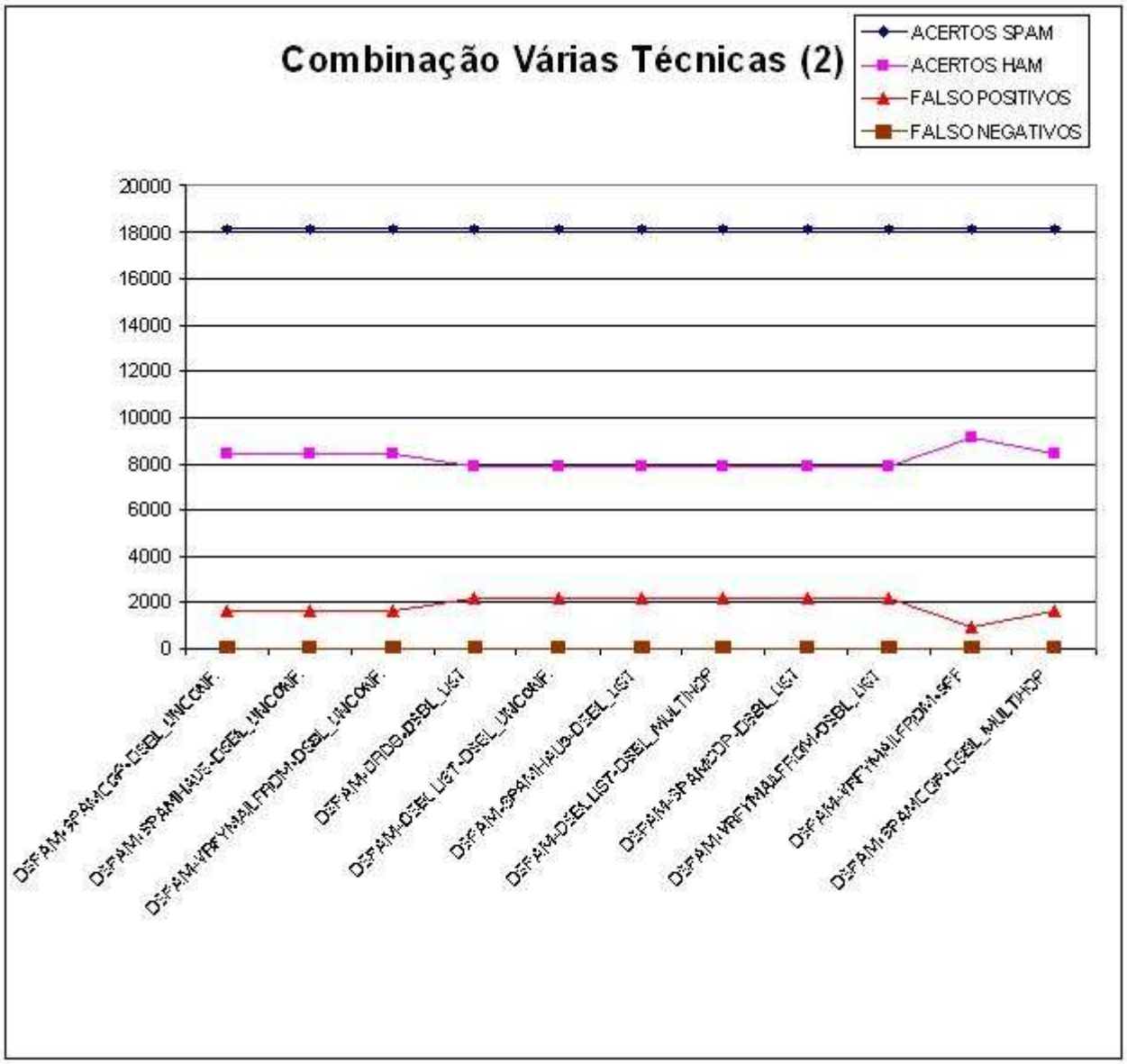

FIGURA 3.15 – Combinação DSPAM + Várias Técnicas - 2

TABELA 3.15 – Combinação DSPAM + Várias Técnicas - 3

Combinação	**Acertos Spam**	**Acertos Ham**	**Falso Positivos**	**Falso Negativos**	**Erros**
DSPAM + ORDB + DSBL Multihop	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + ORDB + Spamcop	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + Spamhaus + DSBL Multihop	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + Spamcop + Spamhaus	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + ORDB + Spamhaus	99,92%	83,62%	5,85%	0,05%	5,90%
DSPAM + Verif. Mailfrom + Spamcop	99,92%	83,54%	5,89%	0,05%	5,94%
DSPAM + ORDB + Verif. Mailfrom	99,92%	83,54%	5,89%	0,05%	5,94%
DSPAM + Verif. Mailfrom + DSBL Multihop	99,92%	83,54%	5,89%	0,05%	5,94%
DSPAM + Verif. Mailfrom + Spamhaus	99,92%	83,54%	5,89%	0,05%	5,94%

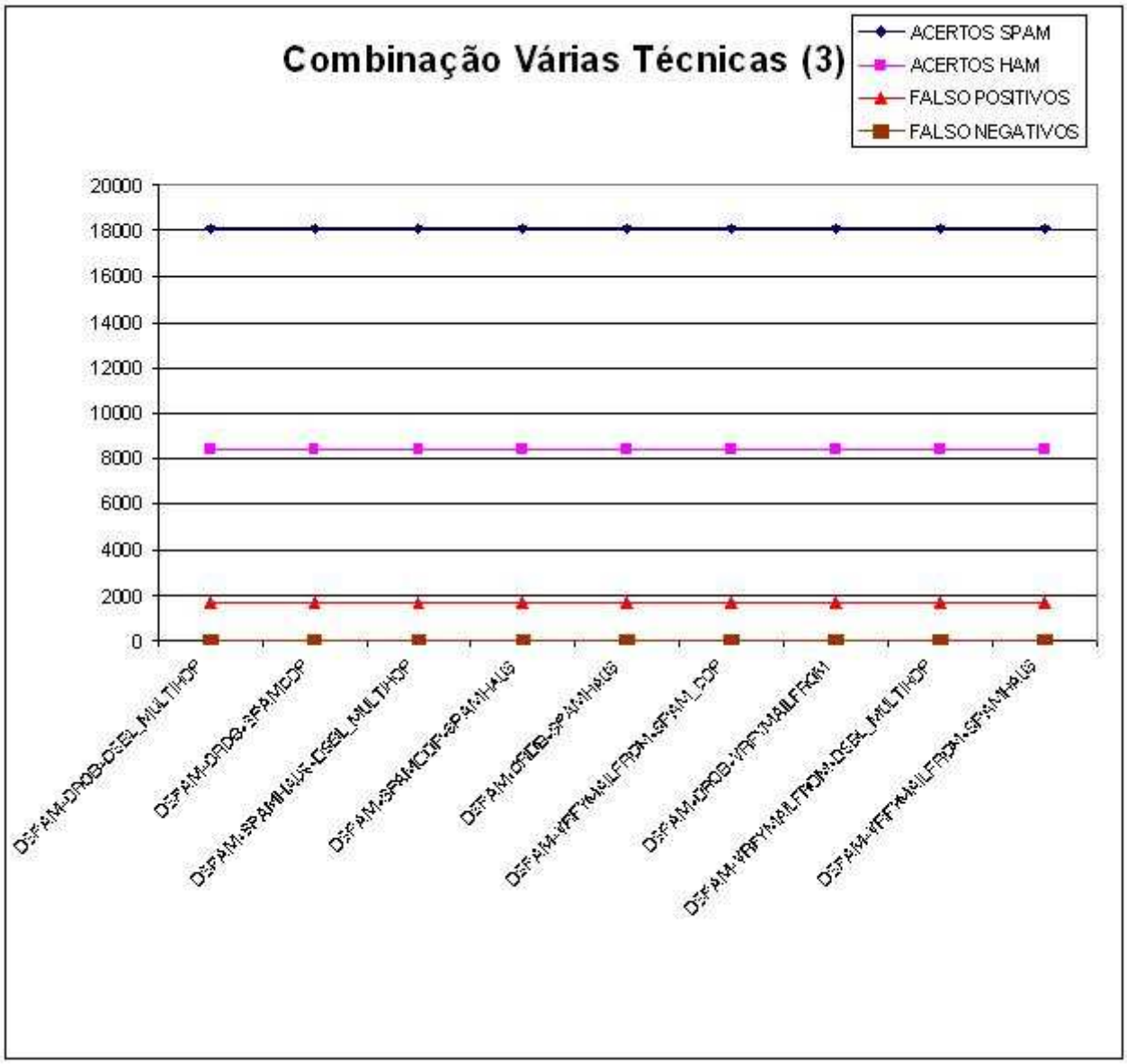

FIGURA 3.16 – Combinação DSPAM + Várias Técnicas - 3

4 Conclusão

Neste trabalho, mais precisamente no capítulo 1, foi feito uma definição do que é spam e sua classificação. Além disso, foi apresentado o histórico do surgimento desse mal moderno, e a extensão do problema que o envio de spam causa no contexto empresarial e doméstico, com uma estimativa de custos mais comuns devido à sobrecarga de mensagens desnecessárias todos os dias. E o problema mais grave, que é a confiabilidade da ferramenta de e-mail para uso no dia a dia das pessoas como ferramenta confiável para comunicação e até mesmo realização de negócios.

Esses problemas estimulam ao desenvolvimento de ferramentas e técnicas cada vez mais avançadas para a classificação de mensagens, contribuindo para um uso mais racional dos recursos de hardware e *link* de internet, e principalmente o aumento de produtividade.

No capítulo 2, foram abordadas diferentes técnicas e ferramentas de código aberto anti-spam mais populares, que são utilizadas para filtragem das mensagens. Algumas ferramentas necessitam de uma base de treinamento, e outras apenas são aplicadas no cabeçalho do e-mail ou na transação SMTP, quando o servidor remetente está enviando ao servidor destinatário. As ferramentas de análise de conteúdo utilizam vários algoritmos e abordagens diferentes para classificação das mensagens.

Uma empresa ou entidade nos dias de hoje, para ter um servidor de correio na internet, obrigatoriamente precisa de técnicas e ferramentas instaladas e configuradas a fim de evitar spam, com o mínimo de falso-positivos. Isso demanda conhecimento técnico avançado, e com todas as alternativas hoje disponíveis, confundem o administrador de correio de qual adotar. O capítulo 3 se dedica justamente a isso, ele procura a melhor combinação possível de técnicas e ferramentas, com o maior grau de acerto na classificação de mensagens, tanto de spams quanto não spams, e ao mesmo tempo expondo as vantagens ou desvantagens na adoção de determinada técnica, como falso-positivos, falso-negaticos, atraso de entrega da mensagem, necessidade de intervenção do administrador, treinamento constante, etc.

O ambiente ideal para tais testes seria um servidor de correio real com alto tráfego de mensagens, porém em virtude dos transtornos que isso poderia causar, foi decidido realizar tais testes em ambiente controlado com bases de mensagens estáticas, impossibilitando tecnicamente os testes de algumas técnicas, como greylisting e tarpit delay, porém os resultados obtidos ainda mantém a sua validade. Como exposto no capítulo 2, essas técnicas também possuem a suas vantagens e desvantagens, sendo a maior desvantagem o atraso da entrega de mensagens legítimas.

Como todos os testes realizados, e com a base de dados dos resultados disponível, chega-se às seguintes conclusões:

- A ferramenta DSPAM foi a que demonstrou ter mais eficiência na detecção de spams em sua configuraçao padrão, porém com um número muito alto de falso-positivos, o que leva a concluir que a necessidade de treinamento constante dessa ferramenta é imprescindível, e o uso inicial deve ser acompanhado de *whitelists*, conforme seu manual de instalação;

- A ferramenta DSPAM, dentre as de análise de conteúdo, foi a que apresentou a alternativa mais interessante para treinamento, já que os usuários finais dos

servidores de e-mail participam da aprendizagem da ferramenta;

- A ferramenta CRM114 também apresentou resultados ótimos de detecção de spam, mas como seu método de treinamento é *Train-on-error*, necessita de muita intervenção do administrador de correio, que precisa separar a base de mensagens classificadas incorretamente para posterior treinamento. Isso até pode ser automatizado através de *scripts*, porém depende ainda da imaginação e conhecimento técnico de quem está implementando a ferramenta;

- A ferramenta Bogofilter também apresentou bons resultados, principalemente na detecção de mensagens não-spam, gerando poucos falso-positivos. Apesar disso, da mesma maneira que a ferramenta CRM114, necessita de muita intervenção do administrador para constante treinemanto da ferramenta;

- Dentre as ferramentas de análise de conteúdo, a que teve o pior resultado foi o SpamAssassin. Isso devido ao número de regras utilizadas, que foi obtido na internet para detecção de spams também na língua portuguesa. O incremento de número de regras no SpamAssassin se mostrou verdadeiros devoradores de CPU, suas expressões regulares são em modo texto, executados através da linguagem Perl, e a atualização e manutenção dessas regras demandam muito conhecimento do técnico em expressões regulares. Nos testes não foi utilizado a base de dados do Razor, em razão de não vir por padrão no SpamAssassin;

- A técnica de verificação do DNS reverso do endereço IP do servidor remetente se mostrou uma ferramenta muito eficiente para detecção de spam, devido ao seu baixo uso de recursos do servidor destinatário. Apenas com uma consulta DNS, pode-se evitar 30,93% dos spams, com apenas 0,73% de falso-positivos, que na verdade são servidores de correio mal configurados. Todo servidor de correio de uma empresa séria tem seu DNS reverso configurado;

- A técnica de verificação do remetente diminui em 7,47% o número de spams, número maior até do que consultas em listas negras;

- O SPF apresentou ainda um desempenho baixo para detecção de spam, mas com sua popularização, esse índice está aumentando. A sua recente adoção em grandes provedores de internet, e sua aprovação pela IETF irão aumentar o nível de acertos na classificação de mensagens, e auxiliar outras ferramentas na classificação;

- Ainda quanto ao SPF, foram encontradas ocorrências de spam em servidores com SPF configurado, ou seja, são domínios não forjados de remetentes legítimos enviando spam. Essa identificação irá ajudar no grau de acerto do uso de *blacklists*;

- Quanto às listas negras públicas, dois fatos comprovados: hoje em dia a sua implantação já não é mais tão eficiente, e apesar de nos testes ter apresentado um nível baixo de falso-positivos, a sua característica de não dar poder ao administrador decidir quais servidores ou domínios são spammers ou não, obriga o uso de *whitelists* customizados;

- Na combinação de duas técnicas, a que apresentou melhores resultados foi o uso de DSPAM junto com SPF, com o grau de acerto de spams da ferramenta DSPAM e o acerto de não-spams da técnica SPF, diminuindo o número de falso positivos;

- Na combinação de várias técnicas, a que apresentou melhor resultado foi o uso de DSPAM combinado com verificação do DNS reverso, verificação do endereço do remetente e SPF. Essas técnicas juntas tiveram o acerto de spams de 99,95%, e falso-negativos de 0,03%. Ainda existe a possibilidade de falso-positivos mesmo nessa melhor combinação encontrada, por isso mesmo que a mensagem seja classificada como spam, deve ser entregue ao usuário para ele poder decidir se aquela mensagem é realmente spam ou não, e ter a alternativa de alterar a classificação de uma mensagem, como o DSPAM proporciona. Apenas nas verificações de transações SMTP a mensagem pode ser descartada se ela for classificada como spam;

- Mesmo a combinação de todas as listas negras públicas, o índice de acerto de spam foi muito baixo. Previsão de extinção dessas listas com o uso das técnicas de SPF e domainkeys para validação da origem? Na verdade, essas listas irão se tranformar em listas de reputação, onde cada domínio terá uma reputação de ser um spammer conhecido ou não, mas o uso dessa lista também obrigará o administrador a usar *whitelists*, pois a decisão de quem irá fazer parte dessas listas de reputação também será dos donos de cada lista, e quem garante que podemos confiar cegamente nessas pessoas ou entidades ?

- A técnica DomainKeys necessita de muito poder de processamento em servidores de alto tráfego e isso pode ser um impedimento de sua popularização;

- A técnica DSPAM proporciona separação da classificação de mensagens por usuário: o que pode ser spam para uma pessoa pode ser importante para outra, a decisão está nas mãos do usuário;

- Se DomainKeys fosse adotado por todos, a confiabilidade do uso de e-mail alcançaria valores elevadíssimos, e seria praticamente impossível aplicar golpes via e-mail como os que existem hoje em dia. Pode ser que DomainKeys seja muito avançado para a época atual, mas no futuro com o poder de processamento das máquinas muito maior, se tornará obrigatório;

Visto todas essas afirmações, conclui-se que as ferramentas e técnica anti-spam estão se tornando cada vez mais avançadas, mas ainda estão um passo atrás dos spammers. Muito foi aprendido nas tentativas de impedir o crescimento do fenômeno do spam, e desse aprendizado surgiram muitas soluções e algoritmos criativos. A implantação dessas ferramentas irão ajudar muito na redução do spam, mas não eliminarão o problema por completo. O problema está na raiz do protocolo SMTP, que quando foi projetado não se preocupou com a autenticidade das mensagens e controle de envio, o que hoje está tentando ser contornado através de SPF e DomainKeys. A garantia de autenticidade da mensagem é a chave para a busca da solução definitiva do problema do spam. E a partir do momento que se tornar muito custoso o envio de mensagens em massa e a identificação e punição das pessoas que enviam spam, o spammer não terá a motivação que tem hoje para o envio de mensagens não solicitadas.

4.1 Trabalhos Futuros

A continuidade desse trabalho pode envolver um estudo mais detalhado de cada ferramenta abordada e de seus algoritmos, fazendo uma analogia entre elas.

Poderia ser também feito, um estudo do uso de recursos de hardware de cada técnica e ferramenta, para estimar o limite máximo de sua implantação.

A aplicação dos testes em um servidor de correio real permitiria analisar as técnicas não testadas, como greylisting e tarpit delay.

O uso de uma alternativa ao protocolo SMTP que garantisse um maior controle da autenticidade da mensagem seria interessante e poderia dar origem à uma nova proposta de funcionamento do protocolo SMTP.

Bibliografia

[APA 2004] The Apache Software Foundation. **Apache License, Version 2.0**. Disponível em <http://www.apache.org/licenses/LICENSE-2.0>.Acesso em: abril/2005.

[API 2003] APIG - All Party Internet Group. **Spam: Report of an Inquiry**. Disponível em <http://www.apig.org.uk/spam_report.pdf>.Acesso em: abril/2005.

[CCE 2004] CCE - Comissão das Comunidades Européias. **Comunicação da Comissão ao Parlamento Europeu, ao Conselho, ao Comitê Econômico e Social Europeu e ao Comitê das Regiões sobre as comunicações comerciais não solicitadas, ou "spam"**. Disponível em <http://europa.eu.int/information_society/topics/ecomm/doc/useful_information/library/communic_reports/spam/pt.pdf>. Acesso em: abril/2005.

[CAU 2005] CAUCE - Coalition Against Unsolicited Commercial Email. **The UCE Problem**. Disponível em <http://www.cauce.org/about/problem.shtml>. Acesso em: abril/2005.

[CEN 2005] JUNKCENTRAL. **The Spam Collection Service**. Disponível em <http://www.vinod.150m.com>. Acesso em: abril/2005.

[CRO 82] CROCKER, David H. **RFC 822 - Standard for the format of ARPA Internet text messages**. Disponível em <http://www.ietf.org/rfc/rfc822.txt>. Acesso em: abril/2005.

[CRO 97] CROCKER, David H. **RFC 2142 - Mailbox Names For Common Services, Roles and Functions**. Disponível em <http://www.ietf.org/rfc/rfc2142.txt>. Acesso em: junho/2005.

[DEN 2003] DENT, Kyle D. **Postfix: The Definitive Guide**. Cambridge, USA: O'Reilly & Associates, 2003. 264p.

[DON 99] DONALDSON, Albert L. **Method and apparatus for filtering junk email**, , US patent 6,321,267. Disponível em <http://www.uspto.gov>. Acesso em: abril/2005.

[FTC 98] FTC - Federal Trade Comission of USA.**FTC Names Its Dirty Dozen: 12 Scams Most Likely to Arrive Via Bulk Email**. Disponível em <http://www.ftc.gov/bcp/conline/pubs/alerts/doznalrt.htm>. Acesso em: abril/2005.

[GEN 2005] Diversos. **Gentoo Linux Web Site**. Disponível em <http://www.gentoo.org>. Acesso em: junho/2005.

[GRA 2002] GRAHAM, Paul. **A Plan for Spam**. Disponível em <http://www.paulgraham.com/spam.html>. Acesso em: abril/2005.

[GRA 2005] GRAHAM, Paul. **The Destiny of Blacklists**. Disponível em <http://www.paulgraham.com/spamhausblacklist.html>. Acesso em: junho/2005.

[GRA 2005] GRAHAM, Paul. **Another SBL Story**. Disponível em <http://www.paulgraham.com/spamhaussbl.html>. Acesso em: junho/2005.

[GRO 2004] GROTE, Marc. **An Overview of the Sender Policy Framework**. Disponível em <http://www.msexchange.org/tutorials/Sender-Policy-Framework.html>. Acesso em: abril/2005.

[HAM 99] HAMBRIDGE, S. **RFC 2635 - A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam*)**. Disponível em <http://www.ietf.org/rfc/rfc2635.txt>. Acesso em: abril/2005.

[HAR 1997] HARKER, Robert. **Selectively Rejecting SPAM Using Sendmail**. Apresentação na Proceedings of the Eleventh Systems Administration Conference (LISA'97), em San Diego, California, Outubro de 1997. Disponível em <http://www.usenix.org/publications/library/proceedings/lisa97/full_papers/22.harker/22.pdf>. Acesso em: abril/2005.

[HAR 2005A] HARRIS, Evan. **Greylisting: The Next Step in the Spam Control War**. Disponível em <http://projects.puremagic.com/greylisting>. Acesso em: abril/2005.

[HAR 2005B] HARRIS, Evan. **Whitelist_ip.txt**. Disponível em <http://cvs.puremagic.com/viewcvs/greylisting/schema/whitelisti_ip.txt>. Acesso em: abril/2005.

[KLE 2001] KLENSIN, J. **RFC 2821 - Simple Mail Transfer Protocol**. Disponível em <http://www.ietf.org/rfc/rfc2821.txt>. Acesso em: abril/2005.

[KNI 2004] KNIGHT, Christopher. **Sender Policy Framework (SPF) - Explained** .Disponível em <http://emailuniverse.com/ezine-tips/?Sender-Policy-Framework-(SPF)—Explained&id=1202&cat=resource>. Acesso em: abril/2005.

[KUB 64] KUBRICK, Stanley. **Dr. Strangelove or How I Learned to Stop Worrying and Love the Bomb**. Disponível em <http://adorocinema.cidadeinternet.com.br/filmes/dr-fantastico/dr-fantastico.htm>. Acesso em: abril/2005.

[LAF 2005] LAFRAIA, Daniel. **SpamBR Rules for SpamAssassin**. Disponível em <http://www.exit0.us/index.php?pagename=BrazilianRules>. Acesso em: junho/2005

[LAN 2001] LANZ, Sylvia. **The rise and fall of corporate electronic mail**. Ubiquity Magazine, Volume 2 Issue 4, Março 2001. Disponível em <http://acm.org/ubiquity/views/s_lanz_1.html>. Acesso em: abril/2005

[LIN 99] LINDBERG, G. **RFC 2505 - Anti-Spam Recommendations for SMTP MTAs**. Disponível em <http://www.ietf.org/rfc/rfc2505.txt>. Acesso em: abril/2005.

[LIN 2003] LINTON, Jeff Connor. **Chi Square Tutorial**. Disponível em <http://www.georgetown.edu/faculty/ballc/webtools/web_chi_tut.html>. Acesso em: abril/2005.

[LOU 2004] LOUIS, Greg. **Greg's Bogofilter Page**. Disponível em <http://www.bgl.nu/bogofilter>. Acesso em: abril/2005.

[MYS 2005] MySQL AB. **MySQL Web Site: The World's Most Popular Open Source Database**. Disponível em <http://www.mysql.com>. Acesso em: junho/2005.

[MOC 87] MOCAPETRIS, P. **RFC 1035 - Domain Names - Implementation and Specification**. Disponível em <http://www.ietf.org/rfc/rfc1035.txt>. Acesso em: abril/2005.

[NIC 2004] NIC BR Security Office, Brazilian Computer Emergency Response Team. **Cartilha de Segurança para Internet Versão 2.0**. Disponível em <http://www.nbso.nic.br/docs/cartilha/cartilha-06-spam.pdf>. Acesso em :abril/2005.

[NIG 96] NIGERIA, The 419 Coalition Website **The Nigerian Scam Defined**. Disponível em <http://home.rica.net/alphae/419coal>. Acesso em: abril/2005.

[ORD 2005] ORDB.org. **ORDB - Open Relay Database Website**. Disponível em <http://www.ordb.org>. Acesso em: maio/2005.

[POL 2004] POLLARD, Jonathan de Boyne. **SPF is harmful. Adopt it**. Disponível em <http://homepages.tesco.net/ J.deBoynePollard/FGA/smtp-spf-is-harmful.html>. Acesso em: abril/2005.

[POS 75] POSTEL, Jonathan B. **RFC 706 - On the Junk Mail Problem**. Disponível em <http://www.ietf.org/rfc/rfc706.txt>. Acesso em: abril/2005.

[POS 82] POSTEL, Jonathan B. **RFC 821 - Simple Mail Tranfer Protocol**. Disponível em <http://www.ietf.org/rfc/rfc821.txt>. Acesso em: abril/2005.

[PRA 2005] PRAKASH, Vipul Ved. **Vipul's Razor**. Disponível em <http://razor.sourceforge.net>. Acesso em abril/2005.

[RAY 2005] RAYMOND, Eric. **Bogofilter Home Page**. Disponível em <http://bogofilter.sourceforge.net>. Acesso em: abril/2005.

[RES 2001] RESNIK, P. **RFC 2822 - Internet Message Format**. Disponível em <http://www.ietf.org/rfc/rfc2822.txt>. Acesso em: abril/2005.

[ROB 2003] ROBINSON, Gary. **A Statistical Approach to the Spam Problem**. Disponível em <http://www.linuxjournal.com/article/6467>. Acesso em: abril/2005.

[SLA 2000] SLASHDOT **MAPS RBL Is Now Censorware**. Disponível em <http://slashdot.org/yro/00/12/13/1853237.shtml>. Acesso em: junho/2005.

[SLE 2005] SLEEPYCAT Software **Berkeley DB Reference Guide**. Disponível em <http://www.sleepycat.com/docs/ref/toc.html>. Acesso em: abril/2005.

[SLT 2005] SLETTNES, Tor. **Spam Filtering for Mail Exchangers**. Disponível em <http://slett.net/spam-filtering-for-mx>. Acesso em: abril/2005.

[SOU 2005] SOUZA, Bruno **An introduction to SPF**. Disponível em <http://linuxfocus.org/English/December2004/article354.shtml>. Acesso em: abril/2005.

[SPA 2005] SPAMHAUS **The Spamhaus Project**. Disponível em <http://www.spamhaus.org>. Acesso em: junho/2005.

[TEI 2004] TEIXEIRA, Renata Cicilini. **Combatendo o Spam**. São Paulo-SP: Novatec, 2004. 171p.

[TEM 2003] TEMPLETON, Brad. **Origin of the term "spam" to mean net abuse**. Disponível em: <http://www.templetons.com/brad/spamterm.html>. Acesso em: abril 2005.

[UTU 2001] UTUG. **Página do grupo de usuários TEX da UFRGS**. Disponível em: <http://gppd.inf.ufrgs.br/~avila/utug>. Acesso em: abril 2005.

[VEN 2004] VENEMA, Wietse. **The Postfix Home Page**. Disponível em <http://www.postfix.org>. Acesso em abril/2005.

[WIK 2005A] WIKIPEDIA. **Bayesian Filter**. Disponível em <http://en.wikipedia.org/wiki/Bayesian_filter>. Acesso em: abril/2005.

[WIK 2005B] WIKIPEDIA. **Sender Policy Framework**. Disponível em <http://en.wikipedia.org/wiki/Sender_Policy_Framework>. Acesso em: abril/2005.

[WIL 2004] WILSON, Ralph F. **SPF Helps Legitimate E-Mail Get through Spam Filters**. Disponível em <http://www.wilsonweb.com/wmtp8/spf_howto.htm>. Acesso em: abril/2005.

[WON 2004] WONG, M. **Internet-Draft : Sender Policy Framework: Authorizing Use of Domains in E-MAIL**. Disponível em <http://www.ietf.org/internet-drafts/draft-schlitt-spf-classic-00.txt>. Acesso em: abril/2005.

[YAH 2004] YAHOO. **DomainKeys: Proving and Protecting Email Sender Identity**. Disponível em <http://antispam.yahoo.com/domainkeys>. Acesso em: junho/2005.

[YER 2002] YERAZUNIS, William S. **Better than Human**. Disponível em <http://www.paulgraham.com/wsy.html>. Acesso em: abril/2005.

[YER 2005] YERAZUNIS, William S. **CRM114 - the Controllable Regex Mutilator**. Disponível em <http://crm114.sourceforge.net>. Acesso em: abril/2005.

[ZDZ 2005] ZDZIARSKI, Jonathan **Nuclear Elephant: The DSPAM Project**. Disponível em <http://www.nuclearelephant.com/projects/dspam>. Acesso em: junho/2005.

[ZYT 2005] ZYTRAX. **HOWTO - Define an SPF Record**. Disponível em <http://www.zytrax.com/books/dns/ch9/spf.html>. Acesso em: abril/2005.

Anexo 1 Scripts e Consultas SQL Utilizados

A.1 Scripts para Treinamento das Ferramentas

A.1.1 Bogofilter

```
#!/bin/bash
for MENSAGEM in $(find work/MENSAGENS/TREINOS/SPAM -print); do
echo "Processando mensagem ${MENSAGEM}"
bogofilter -s <${MENSAGEM}
done
```

A.1.2 CRM114

```
#!/bin/bash
#RODAR COMO ROOT
cd /usr/share/crm114
cssutil -b -r spam.css
cssutil -b -r nonspam.css
for ARQ in $(find work/MENSAGENS/TREINOS/SPAM -print); do
echo "================================="
echo $ARQ
./mailfilter.crm –learnspam –force < ${ARQ} >>/tmp/crm.log
done
for ARQ in $(find work/MENSAGENS/TREINOS/HAM -print); do
echo "================================="
echo $ARQ
./mailfilter.crm –learnnonspam –force < ${ARQ} >>/tmp/crm.log
done
```

A.1.3 DSPAM

```
#!/bin/bash
for ARQ in $(find work/MENSAGENS/TREINOS/SPAM -print); do
echo "================================="
echo $ARQ
dspam –user 'jczucco' –class=spam –source=corpus –stdout < ${ARQ}
done
for ARQ in $(find work/MENSAGENS/TREINOS/HAM -print); do
echo "================================="
echo $ARQ
dspam –user 'jczucco' –class=innocent –source=corpus –stdout < ${ARQ}
done
```

A.2 Scripts de Testes

A.2.1 Script para Catalogar as Mensagens na Base de Resultados

```
#!/bin/bash
MYSQL=”localhost”
USER=”root”
PASS=
# Cadastra as mensagens SPAM for MENSAGEM in $(find SPAM -print -mindepth 2); do
echo ${MENSAGEM}
mysql -u ${USER} -p${PASS} spam -e ”INSERT INTO MENSAGENS (CODIGO, CAMINHO, SPAMASSASSIN, BOGOFILTER, CRM114, ORDB, SPAM_COP, SPAM_HAUS, DSBL_LIST, DSBL_MULTIHOP, DSBL_UNCONFIRMED, RFCIGNORANT, SPF, DOMAINKEYS, GREYLISTING, VRFYHELO, VRFYREVERSO, VRFYMAILFROM, VRFYRCPTO, TARPITDELAY, SPAM_HAM, ASSUNTO, MAIL_FROM, IP_FROM) VALUES (NULL, ’${MENSAGEM}’, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ’S’, NULL, NULL, ”);”
done
#cadastra as mensagens HAM
for MENSAGEM in $(find HAM -print -mindepth 2); do
echo ${MENSAGEM}
mysql -u ${USER} -p${PASS} spam -e ”INSERT INTO MENSAGENS (CODIGO, CAMINHO, SPAMASSASSIN, BOGOFILTER, CRM114, ORDB, SPAM_COP, SPAM_HAUS, DSBL_LIST, DSBL_MULTIHOP, DSBL_UNCONFIRMED, RFCIGNORANT, SPF,DOMAINKEYS, GREYLISTING, VRFYHELO, VRFYREVERSO, VRFYMAILFROM, VRFYRCPTO, TARPITDELAY, SPAM_HAM, ASSUNTO, MAIL_FROM, IP_FROM) VALUES (NULL, ’${MENSAGEM}’, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ”, ’H’, NULL, NULL, ”);”
done
```

A.2.2 Script que Realiza Todos os Testes e Armazena o Resultado em Banco de Dados

```
#!/bin/bash
MYSQ=“localhost”
USER=“root“
PASS=” “
PREFIX=”work/MENSAGENS“
function dominio_existe()
{
DOMINIO=${1}
TESTE_DOMINIO=$(host -t ns ${DOMINIO} — grep -v ”not found: 3(NX-
DOMAIN)”)
[ ”${TESTE_DOMINIO}”] && {
echo EXISTE
} —— {
echo NAOEXISTE
}
}
```

```
function valida_email()
{
TESTE=$(echo ${1} — sed -e ’s/ˆ
+@.
+
..
+/OK/g’)
[ ${TESTE} == ”OK”] && {
echo ”OK”
} —— {
echo ”NOK”
}
}
```

```
function vrfy_dspam()
{
TESTE=$(dspam –user ’jczucco’ –classify –stdout < ${1} — grep ”ˆX-DSPAM-
Result: jczucco; result=
”Spam
)
[ ”${TESTE}”] && {
echo ”NOK”
} —— {
echo ”OK”
}
}
```

```
function vrfy_spamassassin()
{
TESTE=$(spamassassin -t < ${1} — grep ”ˆX-Spam-Status: No,”)
[ ”${TESTE}”] && {
echo ”OK”
} —— {
echo ”NOK”
}
}
```

```
function vrfy_bogofilter()
{
TESTE=$(bogofilter -v < ${1} — grep ”ˆX-Bogosity: No,”)
[ ”${TESTE}”] && {
echo ”OK”
} —— {
echo ”NOK”
}
}
```

```
function vrfy_crm114()
{
cd /usr/share/crm114
TESTE=$(./mailfilter.crm < ${PREFIX}/${1} — grep ”ˆX-CRM114-Status:
SPAM”)
[ ”${TESTE}”] && {
echo NOK
} —— {
echo OK
}
}
```

```
function check_dns_reverso_ip()
{
TESTE=$(host ${1} — grep ”not found: 3(NXDOMAIN)”)
[ ”${TESTE}”] && {
echo NOK
} —— {
echo OK
}
}
```

```
function check_public_list()
{
TESTE=$(host ${1}${2} — grep -v "not found: 3(NXDOMAIN)")
[ "${TESTE}"] && {
echo NOK
} —— {
echo OK
}
}
function check_spf()
{
ENTRADA_SPF=$(host -t txt ${1} — grep spf)
[ "${ENTRADA_SPF}"] && {
TESTE=$(echo ${ENTRADA_SPF} — grep ${2})
[ "${TESTE}"] && {
echo OK_SPF
} —— {
echo NAO_AUTORIZADO_SPF
}
} —— {
echo SEM_SPF
}
}
```

```
[ "${DOMINIO}"] && {
TESTE_DOMINIO=$(dominio_existe ${DOMINIO})
} —— {
TESTE_DOMINIO="NAOEXISTE"
}
[ "${EMAIL_FROM}"] && {
VALIDA_EMAIL_FROM=$(valida_email ${EMAIL_FROM})
} —— {
VALIDA_EMAIL_FROM="NOK"
}
[ "${TO}"] && {
VALIDA_EMAIL_TO=$(valida_email ${TO})
} —— {
VALIDA_EMAIL_TO="NOK"
}
```

```
[ "${CHECK_IP}"] && {
DNS_REVERSO_IP=$(check_dns_reverso_ip ${IP})
# Reorganiza os octetos do IP na ordem da query DNS
OCT1=$(echo -en ${IP} — cut -d'.' -f1)
OCT2=$(echo -en ${IP} — cut -d'.' -f2)
OCT3=$(echo -en ${IP} — cut -d'.' -f3)
OCT4=$(echo -en ${IP} — cut -d'.' -f4)
REV_DNS=$(echo ${OCT4}.${OCT3}.${OCT2}.${OCT1})
CHECK_ORDB=$(check_public_list ${REV_DNS} '.relays.ordb.org' )
CHECK_SPAM_COP=$(check_public_list ${REV_DNS} '.bl.spamcop.net' )
CHECK_SPAMHAUS=$(check_public_list ${REV_DNS} '.sbl-
xbl.spamhaus.org' )
CHECK_DSBL_LIST=$(check_public_list ${REV_DNS} '.list.dsbl.org' )
CHECK_DSBL_MULTIHOP=$(check_public_list ${REV_DNS}
'.multihop.dsbl.org' )
CHECK_DSBL_UNCONFIRMED=$(check_public_list ${REV_DNS} '.uncon-
firmed.dsbl.org' )
CHECK_SPF=$(check_spf "${DOMINIO}${IP}")
} —— {
CHECK_IP="NOK"
DNS_REVERSO_IP="NOK"
CHECK_ORDB="OK"
CHECK_SPAM_COP="OK"
CHECK_SPAMHAUS="OK"
CHECK_DSBL_LIST="OK"
CHECK_DSBL_MULTIHOP="OK"
CHECK_DSBL_UNCONFIRMED="OK"
CHECK_SPF="SEM_SPF"
}
```

```
SPAMASSASSIN=$(vrfy_spamassassin ${1})
BOGOFILTER=$(vrfy_bogofilter ${1})
CRM114=$(vrfy_crm114 ${1})
DSPAM=$(vrfy_dspam ${1})
echo "============================"
echo "ARQUIVO: ${1}"
echo "FROM=${FROM}"
echo "EMAIL_FROM=${EMAIL_FROM}"
echo "VALIDA_EMAIL_FROM=${VALIDA_EMAIL_FROM}"
echo "DOMINIO=${DOMINIO}"
echo "DOMINIO_EXISTE=${TESTE_DOMINIO}"
echo "IP=${IP} --> ${CHECK_IP}"
echo "IP possui DNS reverso=${DNS_REVERSO_IP}"
echo "REV_DNS=${REV_DNS}"
echo "CHECK_ORDB=${CHECK_ORDB}"
echo "CHECK_SPAM_COP=${CHECK_SPAM_COP}"
echo "CHECK_SPAMHAUS=${CHECK_SPAMHAUS}"
echo "CHECK_DSBL_LIST=${CHECK_DSBL_LIST}"
echo "CHECK_DSBL_MULTIHOP=${CHECK_DSBL_MULTIHOP}"
echo "CHECK_DSBL_UNCONFIRMED=${CHECK_DSBL_UNCONFIRMED}"
echo "CHECK_SPF=${CHECK_SPF}"
echo "SPAMASSASSIN=${SPAMASSASSIN}"
echo "BOGOFILTER=${BOGOFILTER}"
echo "CRM114=${CRM114}"
echo "================================"
```

```
# Atualiza o resultado no banco de dados
mysql -u ${USER} -p${PASS} spam -e "UPDATE MENSAGENS SET
SPAMASSASSIN='${SPAMASSASSIN}',
BOGOFILTER='${BOGOFILTER}',
CRM114='${CRM114}', DSPAM='${DSPAM}',
ORDB='${CHECK_ORDB}',
SPAM_COP='${CHECK_SPAM_COP}',
SPAM_HAUS='${CHECK_SPAMHAUS}',
DSBL_LIST='${CHECK_DSBL_LIST}',
DSBL_MULTIHOP='${CHECK_DSBL_MULTIHOP}',
DSBL_UNCONFIRMED='${CHECK_DSBL_UNCONFIRMED}',
RFCIGNORANT='',SPF='${CHECK_SPF}', DOMAINKEYS='',
GREYLISTING='', VRFYHELO='',
VRFYREVERSO='${DNS_REVERSO_IP}',
VRFYMAILFROM='${VALIDA_EMAIL_FROM}',
VRFYRCPTO='',TARPITDELAY='', ASSUNTO='',
MAIL_FROM='${EMAIL_FROM}', IP_FROM='${IP}'
WHERE CAMINHO='${1}';"
```

A.2.3 Scripts para Combinar Duas Tecnicas nas Consultas a Base de Dados

```
#!/bin/bash
USER="root"
PASS=
for ANALISE1 in DSPAM CRM114 BOGOFILTER SPAMASSASSIN; do
for ANALISE2 in ORDB VRFYREVERSO VRFYMAILFROM SPAM_COP
SPAM_HAUS DSBL_LIST DSBL_MULTIHOP DSBL_UNCONFIRMED; do
ACERTO_SPAM=$(mysql -u ${USER} -p${PASS} spam -e "SELECT
count(*) FROM MENSAGENS where ((${ANALISE1}='NOK' and
${ANALISE2}='NOK') or (${ANALISE1}='OK' and ${ANALISE2}='NOK')
or (${ANALISE1}='NOK' and ${ANALISE2}='OK')) and SPAM_HAM='S'")
ACERTO_HAM=$(mysql -u ${USER} -p${PASS} spam -e "SELECT count(*)
FROM MENSAGENS where (${ANALISE1}='OK' and ${ANALISE2}='OK')
and SPAM_HAM='H'")
FALSO_POSITIVOS=$(mysql -u ${USER} -p${PASS} spam -e "SE-
LECT count(*) FROM MENSAGENS where ((${ANALISE1}='NOK' and
${ANALISE2}='NOK') or (${ANALISE1}='OK' and ${ANALISE2}='NOK')
or (${ANALISE1}='NOK' and ${ANALISE2}='OK')) and SPAM_HAM='H'")
FALSO_NEGATIVOS=$(mysql -u ${USER} -p${PASS} spam -e "SE-
LECT count(*) FROM MENSAGENS where (${ANALISE1}='OK' and
${ANALISE2}='OK') and SPAM_HAM='S'")
echo "ANALISE1=${ANALISE1} ANALISE2=${ANALISE2}"
echo "ACERTO_SPAM=${ACERTO_SPAM}"
echo "ACERTO_HAM=${ACERTO_HAM}"
echo "FALSO_POSITIVOS=${FALSO_POSITIVOS}"
echo "FALSO_NEGATIVOS=${FALSO_NEGATIVOS}"
echo
echo "Pressione uma tecla para continuar..."
read a
done
done
```

A.2.4 Scripts para Combinar Tres Tecnicas nas Consultas a Base de Dados

```
#!/bin/bash
USER="root"
PASS=
for ANALISE1 in DSPAM CRM114 BOGOFILTER SPAMASSASSIN; do
for ANALISE2 in ORDB VRFYREVERSO VRFYMAILFROM SPAM_COP
SPAM_HAUS DSBL_LIST DSBL_MULTIHOP DSBL_UNCONFIRMED; do
for ANALISE3 in ORDB VRFYREVERSO VRFYMAILFROM SPAM_COP
SPAM_HAUS DSBL_LIST DSBL_MULTIHOP DSBL_UNCONFIRMED; do
ACERTO_SPAM=$(mysql -u ${USER} -p${PASS} spam -e "SELECT
count(*) FROM MENSAGENS where ((${ANALISE1}='NOK'
and ${ANALISE2}='NOK' and ${ANALISE3}='NOK') or
(${ANALISE1}='NOK' and ${ANALISE2}='NOK' and ${ANALISE3}='OK'
) or (${ANALISE1}='NOK' and ${ANALISE2}='OK' and
${ANALISE3}='NOK' ) or (${ANALISE1}='NOK' and ${ANALISE2}='OK'
and ${ANALISE3}='OK' ) or (${ANALISE1}='OK' and
${ANALISE2}='NOK' and ${ANALISE3}='NOK' ) or (${ANALISE1}='OK'
and ${ANALISE2}='NOK' and ${ANALISE3}='OK' ) or
(${ANALISE1}='OK' and ${ANALISE2}='OK' and ${ANALISE3}='NOK'
)) and SPAM_HAM='S'")
ACERTO_HAM=$(mysql -u ${USER} -p${PASS} spam -e "SELECT count(*)
FROM MENSAGENS where (${ANALISE1}='OK' and ${ANALISE2}='OK'
and ${ANALISE3}='OK') and SPAM_HAM='H'")
FALSO_POSITIVOS=$(mysql -u ${USER} -p${PASS} spam -e
"SELECT count(*) FROM MENSAGENS where ((${ANALISE1}='NOK'
and ${ANALISE2}='NOK' and ${ANALISE3}='NOK') or
(${ANALISE1}='NOK' and ${ANALISE2}='NOK' and ${ANALISE3}='OK'
) or (${ANALISE1}='NOK' and ${ANALISE2}='OK' and
${ANALISE3}='NOK' ) or (${ANALISE1}='NOK' and ${ANALISE2}='OK'
and ${ANALISE3}='OK' ) or (${ANALISE1}='OK' and
${ANALISE2}='NOK' and ${ANALISE3}='NOK' ) or (${ANALISE1}='OK'
and ${ANALISE2}='NOK' and ${ANALISE3}='OK' ) or
(${ANALISE1}='OK' and ${ANALISE2}='OK' and ${ANALISE3}='NOK'
)) and SPAM_HAM='H'")
FALSO_NEGATIVOS=$(mysql -u ${USER} -p${PASS} spam -e "SE-
LECT count(*) FROM MENSAGENS where (${ANALISE1}='OK' and
${ANALISE2}='OK' and ${ANALISE3}='OK') and SPAM_HAM='S'")
echo "ANALISE1=${ANALISE1} ANALISE2=${ANALISE2}
ANALISE3=${ANALISE3}"
echo "ACERTO_SPAM=${ACERTO_SPAM}"
echo "ACERTO_HAM=${ACERTO_HAM}"
echo "FALSO_POSITIVOS=${FALSO_POSITIVOS}"
echo "FALSO_NEGATIVOS=${FALSO_NEGATIVOS}"
echo
echo "Pressione uma tecla para continuar..."
read a
done
done
done
```

A.2.5 Scripts para Combinar Tres Tecnicas nas Consultas a Base de Dados Usando SPF

```
#!/bin/bash
USER="root"
PASS=
for ANALISE1 in DSPAM; do
for ANALISE2 in VRFYREVERSO VRFYMAILFROM SPAM_COP
SPAM_HAUS DSBL_LIST; do
for ANALISE3 in SPF; do
ACERTO_SPAM=$(mysql -u ${USER} -p${PASS} spam -e "SELECT
count(*) FROM MENSAGENS where ((${ANALISE1}='NOK'
and ${ANALISE2}='NOK' and ${ANALISE3}='NAO_AUTORI')
or (${ANALISE1}='NOK' and ${ANALISE2}='NOK' and
(${ANALISE3}='OK_SPF' or ${ANALISE3}='SEM_SPF'))
or (${ANALISE1}='NOK' and ${ANALISE2}='OK' and
${ANALISE3}='NAO_AUTORI' ) or (${ANALISE1}='NOK'
and ${ANALISE2}='OK' and (${ANALISE3}='OK_SPF' or
${ANALISE3}='SEM_SPF')) or (${ANALISE1}='OK' and
${ANALISE2}='NOK' and ${ANALISE3}='NAO_AUTORI'
) or (${ANALISE1}='OK' and ${ANALISE2}='NOK' and
(${ANALISE3}='OK_SPF' or ${ANALISE3}='SEM_SPF'))
or (${ANALISE1}='OK' and ${ANALISE2}='OK' and
${ANALISE3}='NAO_AUTORI' )) and SPAM_HAM='S'")
ACERTO_HAM=$(mysql -u ${USER} -p${PASS} spam -e "SELECT count(*)
FROM MENSAGENS where (${ANALISE1}='OK' and ${ANALISE2}='OK'
and (${ANALISE3}='OK_SPF' or ${ANALISE3}='SEM_SPF')) and
SPAM_HAM='H'")
FALSO_POSITIVOS=$(mysql -u ${USER} -p${PASS} spam -e
"SELECT count(*) FROM MENSAGENS where ((${ANALISE1}='NOK'
and ${ANALISE2}='NOK' and ${ANALISE3}='NAO_AUTORI')
or (${ANALISE1}='NOK' and ${ANALISE2}='NOK' and
(${ANALISE3}='OK_SPF' or ${ANALISE3}='SEM_SPF'))
or (${ANALISE1}='NOK' and ${ANALISE2}='OK' and
${ANALISE3}='NAO_AUTORI' ) or (${ANALISE1}='NOK'
and ${ANALISE2}='OK' and ${ANALISE3}='OK_SPF') or
(${ANALISE1}='OK' and ${ANALISE2}='NOK' and ${ANALISE3}='NOK'
) or (${ANALISE1}='OK' and ${ANALISE2}='NOK' and
${ANALISE3}='OK' ) or (${ANALISE1}='OK' and ${ANALISE2}='OK'
and ${ANALISE3}='NAO_AUTORI' )) and SPAM_HAM='H'")
FALSO_NEGATIVOS=$(mysql -u ${USER} -p${PASS} spam -e "SE-
LECT count(*) FROM MENSAGENS where (${ANALISE1}='OK' and
${ANALISE2}='OK' and ${ANALISE3}='OK') and SPAM_HAM='S'")
echo "ANALISE1=${ANALISE1} ANALISE2=${ANALISE2}
ANALISE3=${ANALISE3}"
echo "ACERTO_SPAM=${ACERTO_SPAM}"
echo "ACERTO_HAM=${ACERTO_HAM}"
echo "FALSO_POSITIVOS=${FALSO_POSITIVOS}"
echo "FALSO_NEGATIVOS=${FALSO_NEGATIVOS}"
echo
echo "Pressione uma tecla para continuar..."
read a
done
```

A.3 Scripts SQL Utilizados

A.3.1 Script SQL que Cria a Base de Resultados dos Testes

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/ spam;
USE spam;
CREATE TABLE MENSAGENS (
CODIGO int(11) NOT NULL auto_increment,
CAMINHO varchar(255) NOT NULL default '',
SPAMASSASSIN varchar(10) NOT NULL default '',
BOGOFILTER varchar(10) NOT NULL default '',
CRM114 varchar(10) NOT NULL default '',
ORDB varchar(10) NOT NULL default '',
SPAM_COP varchar(10) NOT NULL default '',
SPAM_HAUS varchar(10) NOT NULL default '',
DSPAM varchar(10) NOT NULL default '',
DSBL_LIST varchar(100) NOT NULL default '',
DSBL_MULTIHOP varchar(10) NOT NULL default '',
DSBL_UNCONFIRMED varchar(10) NOT NULL default '',
RFCIGNORANT varchar(10) NOT NULL default '',
SPF varchar(10) NOT NULL default '',
DOMAINKEYS varchar(10) NOT NULL default '',
GREYLISTING varchar(10) NOT NULL default '',
VRFYHELO varchar(10) NOT NULL default '',
VRFYREVERSO varchar(10) NOT NULL default '',
VRFYMAILFROM varchar(10) NOT NULL default '',
VRFYRCPTO varchar(10) NOT NULL default '',
TARPITDELAY varchar(10) NOT NULL default '',
SPAM_HAM char(1) NOT NULL default '',
ASSUNTO varchar(255) default '',
MAIL_FROM varchar(255) default '',
IP_FROM varchar(100) NOT NULL default '',
PRIMARY KEY (CODIGO),
KEY idx_SPAM_HAM (SPAM_HAM),
KEY idxCAMINHO (CAMINHO)
) TYPE=MyISAM;
```