UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIAS, GERÊNCIA E SEGURANÇA DE REDES DE COMPUTADORES

DANIEL RICARDO DA SILVA

Sistema para Detecção e Notificação de Problema com DNS Reverso

Trabalho de Conclusão apresentado como requisito parcial para a obtenção do grau de Especialista

Prof. Dr. Rafael Bohrer Ávila
Orientador

Prof. Dr. Sérgio Luis Cechin
Prof. Dr. Luciano Paschoal Gaspary
Coordenadores do Curso

Porto Alegre, dezembro de 2008.

AGRADECIMENTOS

Ao Criador por me disponibilizar a vida, paciência e saúde.

Aos excelentes mestres que nos alimentaram de conhecimentos valiosos, sem os quais a conclusão bem sucedida do curso não seria possível.

Ao meu coordenador Prof. Dr. Rafael Bohrer Ávila, pelas dicas e sugestões valiosas.

À minha família, Bruna e Daíse, pela paciência da espera por mim, nas horas em que me ausentei.

Aos meus pais, Valmor e Neide, por aceitarem minha ausência e me incentivarem incondicionalmente.

Ao Prof. Dr. Luciano Paschoal Gaspary, pelo incentivo e dicas fundamentais.

Aos meus colegas de curso, pela ajuda e contribuição nos trabalhos que, paralelamente, tiveram de ser resolvidos.

Aos meus, familiares, amigos e colegas de trabalho, pela compreensão da minha ausência em alguns momentos.

Aos funcionários da UFRGS que, de uma forma ou de outra, contribuíram para o bom andamento das aulas.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

A	*Address* IPv4
AAAA	*Address* IPv6
BCC	*Blind Carbon Copy*
CC	*Carbon Copy*
CEPTRO	Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações
CERT	Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil
CETIC	Centro de Estudos sobre as Tecnologias da Informação e da Comunicação
CNAME	*Canonical Name*
CPU	*Central Process Unit*
DEC	*Digital Equipment Corporation*
DNS	*Domain Name System*
FQDN	*Fully Qualified Domain Name*
FTP	*File Transfer Protocol*
GB	*Giga Byte*
GUI	*Graphical User Interface*
HINFO	*Host Information*
IMAP	*Internet Message Access Protocol*
IP	*Internet Protocol*
IPv4	*Internet Protocol Version* 4
IPv6	*Internet Protocol Version* 6
ISP	*Internet Service Provider*
LDA	*Local Delivery Agent*
MTA	*Mail Transfer Agent*
MUA	*Mail User Agent*

MX	*Mail Exchange*
NIC	*Network Information Center*
NIC	*Núcleo de Informação e Coordenação*
NS	*Authoritative Name Server*
POP	*Post Office Protocol*
PTR	*Domain Name PoinTeR*
RFC	*Request For Comments*
RR	*Resource Record*
SMTP	*Simple Mail Transfer Protocol*
SMTPD	*Simple Mail Transfer Protocol Daemon*
SOA	*Start Of a Zone of Authority*
SPF	*Sender Policy Framework*
SRV	*Service*
TCP	*Transmission Control Protocol*
TTL	*Time To Live*
TXT	*Text Strings*
UCE	*Unsolicited Commercial E-mail*
UDP	*User Datagram Protocol*
UFRGS	Universidade Federal do Rio Grande do Sul
WKS	*Well Known Service*

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

O propósito desse trabalho é sugerir um modelo de ferramenta, que identifica e, opcionalmente, avisa o administrador do domínio remoto sobre, o problema na configuração de DNS reverso, o qual é um dos maiores responsáveis pela ocorrência de falsos positivos nos sistemas de anti-spam. Dessa forma, foram disponibilizados, no capítulo 2, conceitos que explanam os sistemas envolvidos nesse mecanismo, de maneira a se esperar o bom entendimento da idéia que se propõe. Esses conceitos são referenciais teóricos que abrangem, genericamente, o SPAM, o ANTI-SPAM, o sistema de correio eletrônico (MUA e MTA), o DNS e o DNS Reverso.

Fica a cargo do leitor, a construção de um protótipo, caso haja interesse.

Palavras-Chave: spam, DNS, DNS reverso, anti-spam, MUA, MTA.

System for Detection and Notification of Problem with Reverse DNS

ABSTRACT

This work has the purpose of suggest a type of tool, which identifies, and optionally prompts the administrator of remote domain about problem in setting up of reverse DNS, which is one of the most responsible for the occurrence of false positives in anti-spam systems. Thus, were provided in Chapter 2, concepts that explain the systems involved in this mechanism, so as to be expected the proper understanding of the idea that if suggests. These concepts are the theoretical references that relate generally the SPAM, the ANTI-SPAM, the e-mail system (MUA and MTA), DNS and Reverse DNS.

Is borne by the reader, building a prototype, if there is interest.

Keywords: spam, DNS, reverse DNS, anti-spam, MUA, MTA.

1 INTRODUÇÃO

De acordo com a Symantec (2008), em agosto de 2008, 80% dos e-mails que trafegaram pela Internet, no mundo, foram spams. No mesmo período do ano de 2007 esse número era de 73% (SYMANTEC, 2007). Isso mostra a quantidade de spam aumentando significativamente em detrimento à enorme quantia de dinheiro e talentos investidos contra esse grande transtorno na rede mundial.

Segundo a empresa Marshal (2008) o consenso é que mais de 150 bilhões de mensagens de spams circulam diariamente na Internet. Essas mensagens indesejadas consomem largura de banda e recursos, tempo e dinheiro dos destinatários, além de tornar-se um dos maiores problemas de segurança da Internet, pois os *spammers* têm em suas mãos uma forma de distribuição, em massa, de *malwares*, dentre diversas outras pragas. Ainda, o aumento dos spams tornou-se largamente possível com o advento das *botnets*, que são redes constituídas de milhares de computadores pessoais infectados controlados remotamente pelos “piratas”, de onde se originam cerca de 80% das mensagens de spams no mundo. Além disso, em teoria, um *software* filtro de e-mails, ou equipamento para esse fim, permite uma mensagem legítima passar enquanto bloqueia o spam. Mas o filtro pode se enganar e deixar passar um spam, gerando o que se chama de falso negativo, ou ele pode bloquear um e-mail legítimo denotando um falso positivo (SUN, 2008).

Apesar dos altos investimentos direcionados para uma guerra digital que parece não ter fim, onde de um lado estão os *spammers* e do outro especialistas, empresas, governos, etc. as estatísticas mostram o aumento expressivo da circulação dessas indesejadas mensagens através dos anos. Contudo, não podemos desanimar. Temos a obrigação, como usuários e especialistas, de continuar batalhando para a disseminação de ferramentas e descobertas que possam auxiliar nesse combate, buscando uma estatística mais animadora.

Refletindo sobre esse grande incômodo tecnológico é que propõe-se, nesse trabalho, um sistema para detecção e notificação de problemas com o DNS reverso em servidores de e-mail.

Dentre as técnicas usadas para combater o spam, considera-se a verificação de DNS reverso uma das mais eficazes. Essa técnica

consiste em definir um critério de legitimidade das mensagens enviadas por um servidor com base na configuração de DNS reverso do mesmo. Infelizmente, o uso dessa técnica freqüentemente gera diversos falsos positivos (mensagens legítimas são rejeitadas), devido à grande incidência de servidores mal configurados. Esse problema cria um obstáculo à aceitação do uso desse mecanismo por parte dos usuários. A idéia é oferecer a um administrador de rede uma maneira fácil de alertar outros administradores sobre configurações inadequadas nos servidores que administram. O sistema baseia-se na verificação dos *logs* de um MTA e procura, dentre os servidores que interagiram com o domínio local, por falhas de DNS reverso. Após filtragem segundo critérios de correção, o sistema deve oferecer ao administrador local a possibilidade de notificação dos sistemas remotos.

O restante desse documento está estruturado da seguinte maneira:

- o capítulo 2 é um referencial teórico sobre o spam, o anti-spam, o sistema de correio eletrônico (MUA e MTA), o DNS e o DNS Reverso, com o intuito de ilustrar, basicamente, o funcionamento desses mecanismos;
- no capítulo 3 é apresentada a principal contribuição do trabalho, com a explicação sobre o método proposto, bem como o resultado obtido através de experimentos feitos em *logs* gerados em um servidor de e-mails real;
- finalmente, o capítulo 4 traz as conclusões.

2 RERENCIAL TEÓRICO

Este capítulo tem o objetivo de descrever alguns conceitos dos mecanismos e sistemas mencionados nesse trabalho. As informações aqui apresentadas devem ilustrar o funcionamento básico desses mecanismos objetivando um entendimento genérico para o leitor. Assim, espera-se que o modelo proposto de ferramenta seja compreendido na íntegra.

2.1 Spam

O Cert.br define muito bem o que é spam:

> Spam é o termo usado para se referir aos e-mails não solicitados, que geralmente são enviados para um grande número de pessoas. Quando o conteúdo é exclusivamente comercial , este tipo de mensagem também é referenciada como UCE (do inglês *Unsolicited Commercial E-mail*) (CERT.BR, 2006).

Sabe-se que o spam é extremamente prejudicial para os usuários da rede, sejam esses finais ou empresas, pois traz consigo uma grande quantidade de inconvenientes. Dificilmente exista, no mundo, alguém que nunca tenha recebido uma mensagem indesejada pelo correio eletrônico, embora existam spams via mensagens instantâneas ou redes de relacionamento, também. Essa mensagem não solicitada pode vir constituída de arquivos anexados “maliciosos” como, *worms*, vírus, cavalos de tróia, imagens desagradáveis, etc.; pode vir carregada com *links* para *sites* que possam conter material ilegal ou que possam executar códigos maliciosos ou, ainda, induzir que o usuário faça o *download* de pragas cibernéticas.

Existem muitos outros tipos de mensagens de spam, com a mais variada forma de atingir o maior número possível de pessoas e com uma quantidade imensa de motivos e possibilidades. O limite está na imaginação dos que contribuem para isso, os *spammers*; um grupo constituído de pessoas intencionadas a tirar algum tipo de proveito massificando os spams. A figura 2.1 mostra os principais tipos de spam atuantes na estatística, no segundo trimestre de 2008.

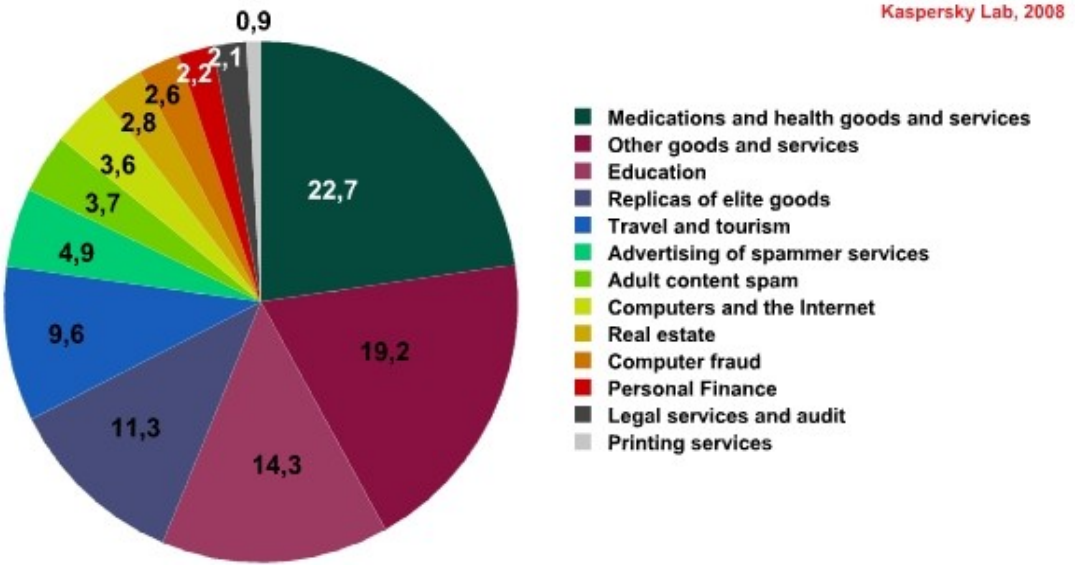

Figura 2.1: Principais tipos de spams trafegando na rede mundial de computadores (GUDKOVA, 2008).

Mas o fato é que, independente do tipo e motivo do spam, os prejuízos são gigantescos. Os usuários perdem tempo e dinheiro recebendo e identificando esses e-mails; os recursos de rede e *backbone* são consumidos; a produtividade dos usuários cai; caixas repletas de spams deixam de receber qualquer outra mensagem, pois atingiram sua cota; etc. E, embora muito tem-se feito para diminuir esse tipo de problema digital, ainda há muito por se fazer; desde a conscientização dos internautas, até a melhoria e criação de mecanismos cada vez mais eficientes. Talvez o mais complicado seja conscientizar os usuários para não clicarem em qualquer *link* ou arquivos suspeitos recebidos por e-mail, além de manterem seus sistemas atualizados, contribuindo para a diminuição das *botnets*, as quais, muitas vezes, aproveitam-se de *exploits* em sistemas desatualizados para a sua expansão.

Como curiosidade, vale a pena mencionar o surgimento do termo “SPAM” conforme consta no site do Antispam.br:

> Boa parte das curiosidades sobre spam diz respeito à origem do termo. Tudo começou com o SPAM® , com letras maiúsculas, um presunto condimentado (***SP****iced h****AM***) e enlatado americano. O SPAM® (www.spam.com) é fabricado pela Hormel Foods (www.hormel.com), desde 1930, e tem uma legião de fãs no mundo inteiro.
>
> O famoso presunto foi tema de uma cena que o eternizaria em um dos programas de TV do grupo de comediantes Monty Python, sempre lembrados por filmes clássicos como: "*Monty Python and the Holy Grail*" e "*Monty Python – The Meaning of Life*".
>
> Numa das cenas do programa "*Monty Python´s Flying Circus TV Show*", um grupo de vikings está em uma taverna, onde entra um casal que consulta o cardápio, cujos pratos são todos feitos com SPAM® . Enquanto o casal conversa com a garçonete, os vikings recitam diversas vezes um texto extremamente chato, repetindo a palavra SPAM® . A frase mais repetida é: "*Spam spam spam spam. Lovely spam!*" (ANTISPAM.BR, 2008).

Existem algumas teorias sobre qual foi a primeira mensagem de spam; no entanto, há duas que podem ser consideradas mais conhecidas. Uma dessas teorias diz que o primeiro spam pode ter sido enviado por Gary Thuerk, funcionário da DEC, no início de maio de 1978, quando enviou uma mensagem com a propaganda do novo computador DEC-20 para 320 (limite do sistema da época) caixas de e-mails da Arpanet, acreditando ser de interesse dos integrantes dessa rede. A outra mensagem, também considerada como o primeiro spam, foi enviada por dois advogados, Cantel e Siegel, em 5 de março de 1994, com um texto sobre uma loteria de *Green Cards* para um grupo na USENET (ANTISPAM.BR). A resposta para esses dois incidentes foi de muita indignação e as reclamações foram muitas, por parte dos receptores.

Independentemente de qual foi a primeira mensagem de spam, o fato é que houve um começo e, desde então, o problema vem aumentando muito com o passar dos anos. Conforme está ilustrado na figura 2.2, vê-se que o Brasil, no terceiro trimestre de 2007, estava em quinto lugar, com responsabilidade no envio de 3,7%, na lista dos 12 países que mais enviaram spam. Em primeiro lugar estava os Estados Unidos com 28,4%, em segundo, bem abaixo, vem a Coréia do Sul com 5,2%, a China vem em terceiro com 4,9%, seguida da Rússia com 4,4%.

Position	Country	Percentage
1	United States	28.4%
2	South Korea	5.2%
3	China (inc.Hong Kong)	4.9%
4	Russia	4.4%
5	Brazil	3.7%
6	France	3.6%
7	Germany	3.4%
8	Turkey	3.2%
9	Poland	2.7%
10	United Kingdom	2.4%
11	Romania	2.3%
12	Mexico	1.9%
Others		33.9%

Figura 2.2: Os 12 países que mais influenciaram nas estatísticas de spams no terceiro trimestre de 2007 (SOPHOS, 2007).

Ainda, vale destacar que o aumento está bastante relacionado, também, com o advento das *botnets*, redes compostas por microcomputadores explorados e controlados remotamente por grupos de piratas digitais. Essas redes também são conhecidas como *zombie army* e, conforme já mencionado, têm responsabilidade no envio de mais de 80% de todos os spams. A figura 2.3 relaciona os países que mais possuem essas redes.

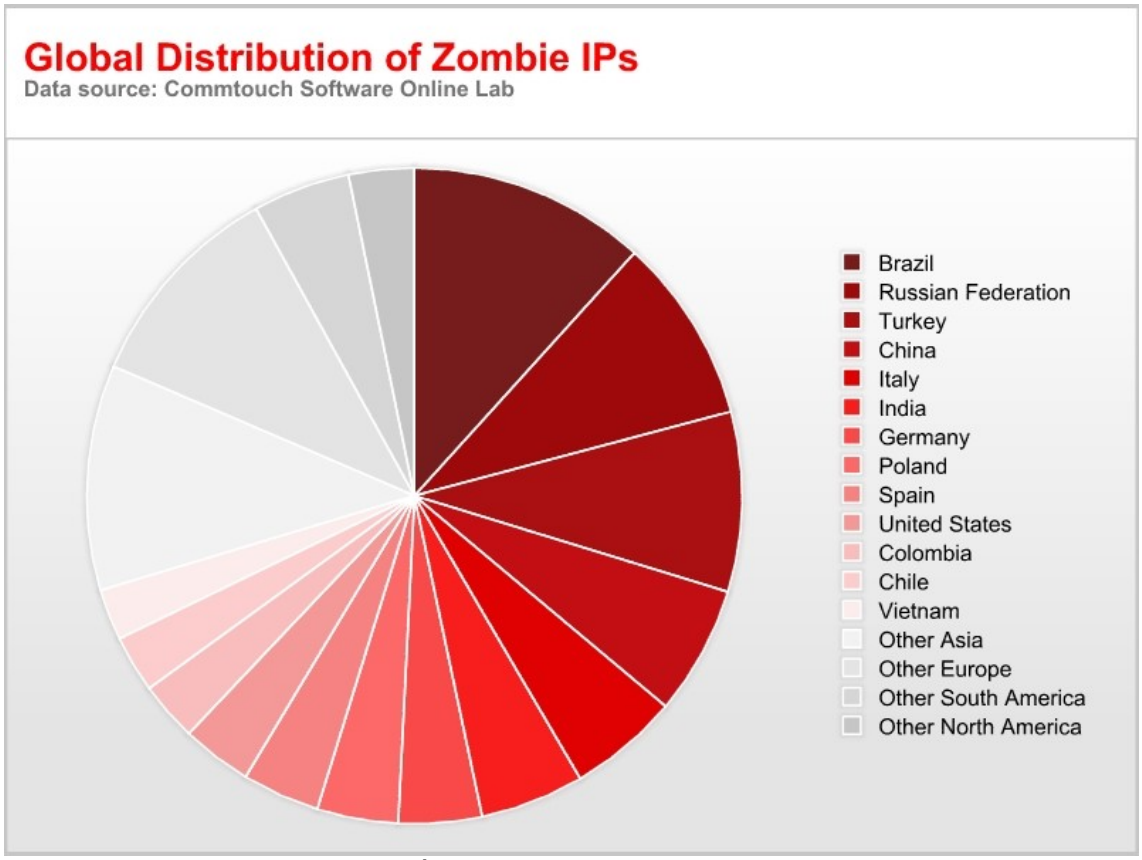

Figura 2.3: Lista dos países que mais possuem redes formadas por computadores zumbi (ZOMBIE, 2008).

Com base nessa problemática do spam, diversas técnicas e ferramentas são desenvolvidas; uma delas é o anti-spam.

2.1.1 Anti-Spam

O anti-spam, como o nome sugere, é o sistema responsável por bloquear (ou filtrar) mensagens indesejadas. Existem diversas formas e ferramentas para dificultar o trabalhos dos *spammers*, mas, certamente, o método mais usado é o filtro de e-mails.

Um sistema de anti-spam, filtro de e-mails, pode ser configurado de diversas maneiras, dependendo da ferramenta e do que se pretende. Basicamente, o anti-spam pode ser instalado no servidor e os filtros são gerenciados por quem administra esse servidor e/ou pelo usuário, dono da conta de e-mail, aplicando suas regras, as quais ficam armazenadas centralizadamente. O anti-spam também pode ser aplicado na estação de trabalho, onde cada usuário é responsável por administrar a ferramenta de anti-spam, conforme vai recebendo seus e-mails, sendo que, dessa forma, as regras são mantidas no *host* local.

Sucintamente, um filtro de anti-spam pode trabalhar com listas negras, listas brancas e quarentena. Assim, nas listas negras o usuário insere endereços de IP, domínios e e-mails daqueles considerados *spammers*. Dessa forma, tudo que for recebido e que case com o conteúdo das listas negras será descartado. Da mesma forma, como nas listas negras, nas listas brancas são colocados

dados de caixas consideradas legítimas, ou seja, das quais se deseja receber e-mails. Com isso, todas as mensagens que chegarem e casarem com as informações nas listas brancas serão recebidas e colocadas na caixa de entrada do usuário. Já o recurso de quarentena, é um local onde são armazenadas possíveis mensagens de spam. A quarentena deve ser verificada com alguma freqüência pelo internauta, pois é lá que o usuário poderá encontrar as mensagens consideradas spam por engano, os chamados falsos positivos (ANTISPAM.BR).

A aplicação de um filtro de spams deve ser bem planejada antes que ela seja implementada em uma infra-estrutura de rede, pois pode gerar algum desconforto inicial pela grande quantidade de falsos negativos e falsos positivos que possam ocorrer.

Os métodos e ferramentas disponíveis possuem suas vantagens e desvantagens. Mas, no final das contas, poderá reduzir bastante a quantidade de e-mails indesejados na estrutura, valendo a pena o seu estudo, sem dúvidas.

2.2 O Sistema de Correio Eletrônico

Basicamente, o sistema de correio eletrônico é composto por agentes. Um desses agentes é responsável pela interação do usuário com as mensagens, ou seja, através desse agente é possível que um usuário abra e leia uma mensagem, responda essa mensagem, crie uma nova mensagem e, ainda, encaminhe uma mensagem recebida para outras contas de e-mail. Esses sistemas, com os quais os usuários podem interagir, são conhecidos como MUA e podem ser bastante avançados, no sentido de possuírem mecanismos gráficos, agenda, catálogo de endereços, módulos para criptografia e assinatura digital, anti-spam integrado, calendário, etc. Exemplos de MUAs gráficos: Mozilla Thunderbird (THUNDERBIRD, 2008), Evolution (EVOLUTION, 2008), Microsoft Outlook (MICROSOFT, 2008), Sylpheed (SYLPHEED, 2008). Ainda, um MUA pode ser bem simplificado visualmente, porém, não menos avançado, focando principalmente na funcionalidade para a qual foi construído e a interação do usuário é feita a partir de comandos ou menus em modo texto. Exemplos de MUAs em modo texto: Mutt (MUTT, 2008), Binmail (BINMAIL, 1998) , Cone (CONE, 2008), Mail (MAIL, 2008), etc. O outro agente possibilita a entrega dos e-mails aos seus destinatários, isto é, esse agente é responsável por transferir essas mensagens, deslocando-as entre a origem e o destino. Esse subsistema, conhecido como MTA, também pode ser chamado de *mail transport agent*, *message transfer agent* ou smtpd (MAIL Transfer, 2008); podendo ser um processo sendo executado em segundo plano, transportando mensagens de correio eletrônico entre o sistema (TANENBAUM, 1997).

2.2.1 MUA

O MUA, conforme já mencionado, é uma aplicação necessária para que o usuário receba e envie mensagens de e-mail. Aceita uma

variada quantidade de comandos, os quais podem ser em modo texto (menos amigável) ou através de janelas gráficas (GUI), compostas por ícones e menus, onde a interação com os comandos exige a utilização de um mouse. No entanto, a funcionalidade é a mesma; enviar e receber mensagens de *mailbox*.

Ao enviar uma mensagem de e-mail o usuário deve passar algumas informações para o *software* cliente de e-mail como, o endereço do destinatário, obviamente de uma forma que o usuário possa lidar, como no formato DNS (caixa_de_email@dominio), opcionalmente o assunto da mensagem e a mensagem em si. Esse usuário pode enviar a mensagem para uma lista de endereços e, ainda, poderá incluir a prioridade e o nível de segurança, além de anexar um determinado arquivo. Ao clicar no botão “Enviar”, ou executar o comando para isso, essa parte do processo está finalizada, isto é, o remetente não precisa fazer mais nada para que a mensagem chegue até o destinatário, pois essa tarefa cabe ao MTA.

Ao executar a aplicação cliente de e-mail, normalmente, ela irá verificar a existência de novas mensagens na caixa de entrada, podendo, em seguida, exibir o número de mensagens novas, o número total de mensagens na *mailbox*, exibir uma mensagem de que não há novos e-mails, mostrar um resumo de cada mensagem; essas características e funcionalidades dependerão muito do *software* utilizado, os quais são inúmeros (TANENBAUM). Na figura 2.4 pode-se notar os principais campos, responsáveis pelo transporte, que devem estar contidos no cabeçalho de uma mensagem.

Cabeçalho	**Significado**
To:	O(s) endereço(s) de correio eletrônico do(s) destinatário(s) principal(s)
Cc:	O(s) endereço(s) de correio eletrônico do(s) destinatário(s) secundário(s)
Bcc:	O(s) endereço(s) de correio eletrônico para cópias carbono cegas
From:	A(s) pessoa(s) que criou(aram) a mensagem
Sender:	O endereço de correio eletrônico do remetente
Received:	A linha que é incluída por cada agente de transferência durante o percurso
Return-Path	Pode ser usada para identificar um caminho de volta ao remetente

Figura 2.4: Os campos do cabeçalho RFC 822 relacionados ao transporte de mensagens (TANENBAUM, 1997).

Os campos que constam acima são bem descritos por Tanenbaum (1997, p. 743-744):

> O campo *To:* indica o endereço DNS do destinatário principal. Também é possível ter vários destinatários. O campo *Cc:* contém os endereços dos destinatários secundários (se houver). Em termos de entrega, não há distinção entre os destinatários principal e secundário. Trata-se de uma diferença inteiramente psicológica importante apenas para as pessoas envolvidas e que não afeta o sistema de correio eletrônico. O termo *Cc:* (*Carbon copy*) já está meio ultrapassado, pois os computadores não utilizam papel carbono, mas já está consagrado. O campo *Bcc:* (*Blind carbon copy*) é semelhante ao campo *Cc:*, a diferença é o fato de que essa linha é eliminada de todas as cópias enviadas aos destinatários principais e secundários. Esse recurso permite que as pessoas enviem cópias a terceiros sem que os destinatários principais e secundários saibam disso.
>
> Os dois campos seguintes, *From:* e *Sender:* informam quem escreveu e enviou a mensagem, respectivamente. Nem sempre esses campos conterão valores iguais. Por exemplo, um executivo pode escrever uma mensagem, mas na verdade sua secretária é quem acaba transmitindo-a. Nesse caso, o executivo seria listado no campo *From*: e a secretária no campo *Sender:*. O campo *From:* é obrigatório, ao passo que *Sender:* pode ser omitido se for igual a *From:*. Esses campos são necessários para o caso de a mensagem ser devolvida ao remetente se não puder ser entregue.
>
> Uma linha contendo *Received:* é incluída por cada agente de transferência de mensagem durante o percurso até o destinatário. A linha contém a identidade do agente, a data e a hora em que a mensagem foi recebida e outras informações que podem ser usadas para localização de *bugs* no sistema de roteamento.
>
> O campo *Return-Path:* é incluído pelo agente de transferência de mensagem e seu objetivo é informar como voltar ao remetente. Na teoria, essas informações podem ser obtidas a partir de todos os cabeçalhos *Received:* (exceto pelo nome da *mailbox* do remetente), mas ele raramente é preenchido dessa forma e, em geral, contém apenas o endereço do remetente.

Além dos campos descritos anteriormente, as mensagens podem conter vários outros campos de cabeçalhos, sobre os quais não serão feitas menções nesse trabalho.

2.2.2 MTA

O MTA é o *software* responsável por transferir uma mensagem de e-mail de um computador para outro, ou seja, do remetente para o destinatário. E é chamado de servidor de e-mail o equipamento no qual roda uma aplicação de MTA. Resumidamente, o MTA recebe as mensagens de e-mail de um outro MTA (*relaying*) ou de um MUA. O MTA é transparente para o usuário, o qual interage com o MUA. Sempre que uma mensagem é recebida pelo MTA, a essa mensagem é adicionado um "*Received:*" no campo do cabeçalho, no início da mensagem. Assim, há uma forma de registrar por quais MTAs essa mensagem passou e em que ordem isso ocorreu (WIKIPEDIA).

A forma mais simples de se enviar um e-mail de um *host* para outro, seria o estabelecimento de uma conexão entre esses dois *hosts*. A realização dessa conexão, feita via protocolo SMTP, seria efetivada entre o MTA do remetente e o MTA do destinatário, baseando-se no campo *To:* do cabeçalho, especificamente no endereço de domínio. Por essa conexão aberta, após o remente ter clicado no botão enviar do MUA (ou um comando caso seja modo texto), o MTA do remetente enviaria comandos em modo texto para o MTA do destinatário, o qual responderia esses comandos, por esse mesmo canal. Esse diálogo seria através do protocolo SMTP, conforme ilustrado na figura 2.5, onde "R" é o remetente e "D" representa o destinatário, baseada no exemplo do site www.tldp.org.

```
R: MAIL FROM:<remetente@remetente.com>
D: 250 OK
R: RCPT TO:<destinatario@destinatario.com>
D: 250 OK
R: DATA
D: 354 Start mail input; end with <CRLF>.<CRLF>
R: From: "Remetente" <remetente@remetente.com>
R: Message-Id: <199711131704.MAA18447@remetente.com>
R: Subject: Você viu meu coelho branco?
R: To: destinatario@destinatario.org (Destinatario)
R: Date: Thu, 13 Nov 1997 12:04:05 -0500 (EST)
R: Content-Type: text
R:
R: Estou muito preocupado. Temo que ele possa ter caído num buraco.
R: --
R:                                                    >>remetente>>
R: .
D: 250 OK
```

Figura 2.5: Representa como seria o diálogo entre o MTA de um remente e o MTA de um destinatário, ligados diretamente via SMTP.

A forma simplificada, conforme descrita no parágrafo anterior, está ilustrada na figura 2.6, retirada do *site* www.tldp.org, onde o autor exemplifica o remetente como Alice e o destinatário como Bob.

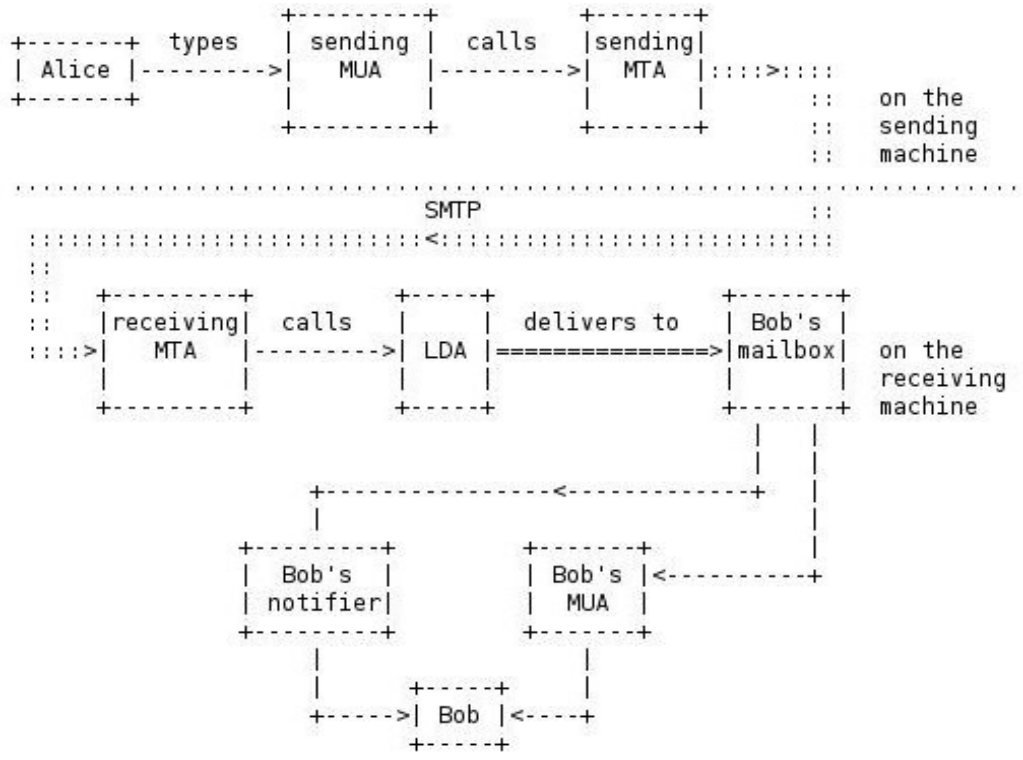

Figura 2.6: Sistema de correio eletrônico simplificado (AZNAR, 2000).

Por fim, o MTA do destinatário teria recebido a mensagem enviada pelo remetente e adicionado o campo "*Received:*" (*Received*: (*from*: remetente@remetente.com)) que serve para registrar por quais MTAs essa mensagem passou e em que ordem isso ocorreu. Na figura 2.6, nota-se a presença do LDA (Agente Local de Entrega) o qual tem a função de receber a mensagem modificada pelo MTA e acrescenta-la na *mailbox* do receptor. Finalmente, o receptor poderá ler suas mensagens recebidas executando o MUA.

No entanto, esse método não é eficiente, pois se o *host* de destino estiver desligado (ou desconectado da rede), o e-mail não poderá ser entregue e, conseqüentemente, a mensagem em questão entraria numa fila no domínio do remetente e tentativas de reenvio seriam realizadas, até que acontecesse a expiração do tempo. Por conseguinte, uma mensagem de notificação seria retornada para o remetente.

Na prática o que acontece é algo diferente, com uma infra-estrutura mais robusta, na qual há um envolvimento maior de mecanismos, protocolos e dispositivos. Tudo para tentar prover garantias no bom funcionamento do sistema de correio eletrônico.

A grande maioria dos usuários atualmente conectam-se na rede mundial através de um ISP (Provedor de Serviços de Internet) e possuem suas contas de e-mail configuradas na infra-estrutura do seu ISP contratado, para a qual são enviadas as mensagens eletrônicas. E, quando o internauta deseja ler essas mensagens, precisa conectar-se ao seu ISP e abrir o MUA para, dessa forma, estabelecer uma conexão e receber essas mensagens na sua máquina local.

É um cenário bastante diferente do mencionado anteriormente, pois, naquele caso, a mensagem era colocada em uma fila e várias tentativas de reenvio eram realizadas, até que o tempo se expirava e a mensagem era retornada; ou era possível estabelecer a comunicação com o *host* do destinatário e a mensagem era enviada. Porém, enquanto a mensagem encontra-se em uma fila, ela não é considerada enviada e fica sujeita ao retorno no caso do tempo expirar. Por outro lado, nesse caso, em que há a presença de um (ou vários) ISP, uma mensagem enviada para os servidores de e-mail localizados na infra-estrutura desse ISP considera-se efetivamente enviada, mesmo que o destinatário ainda não a tenha lido, sendo que essa mensagem pode ficar lá indeterminadamente.

Um protocolo de e-mail remoto torna isso possível, ou seja, possibilita que e-mails sejam mantidos em um servidor para serem consultados/baixados através de uma conexão, por uma aplicação cliente de e-mail. Há dois tipos de protocolos de e-mail remoto mais usados nos dias de hoje, o POP3 que é suportado por todos os ISPs e o IMAP que está sendo adotado cada vez mais pelos provedores de serviços de Internet. Esses dois protocolos são bem similares na

lógica, apesar de usarem comandos diferentes (AZNAR). A figura 2.7 mostra como acontece essa interação.

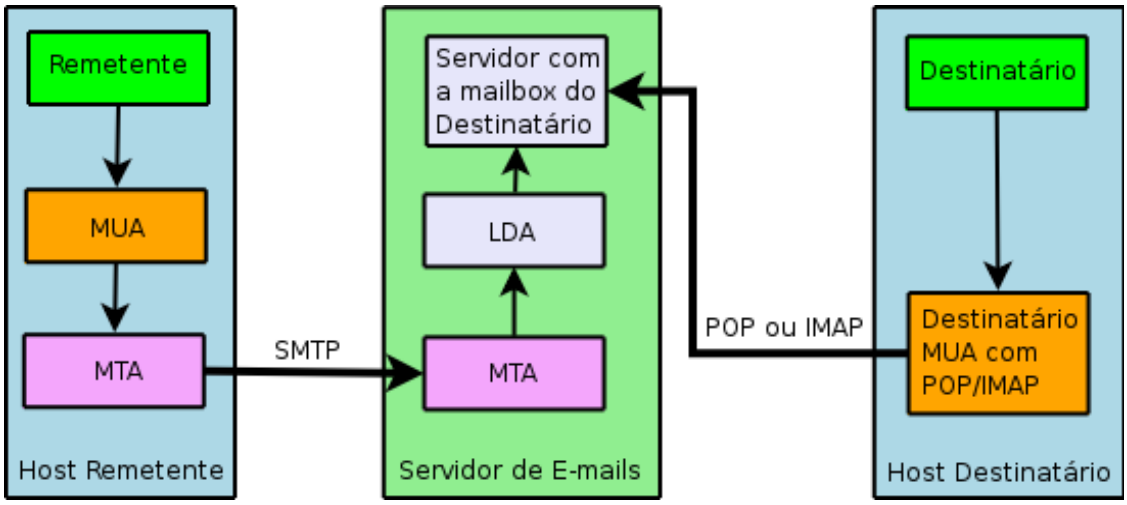

Figura 2.7: Interatividade entre alguém que envia um e-mail, o servidor de e-mails do receptor e o destinatário desse e-mail.

Além de tudo, para funcionar corretamente, a estrutura de correio eletrônico depende de um outro sistema, o DNS. Essa necessidade vem do princípio de que os nomes entendidos pelos seres humanos precisam ser convertidos para números que por sua vez são compreendidos pelos sistemas computacionais, e vice-versa.

2.3 DNS

O incentivo maior para o desenvolvimento do DNS foi o crescimento da Internet. Pois, bem no início, a forma como se mapeava os endereços, com os seus nomes correspondentes de máquinas, era através de um único arquivo em formato texto (hosts.txt), o qual era enviado via FTP para todos os *hosts* em rede. A figura 2.8 mostra uma estrutura com seis *workstations*, cada uma deve conter um arquivo hosts.txt de mapeamento host/IP representado na figura 2.9.

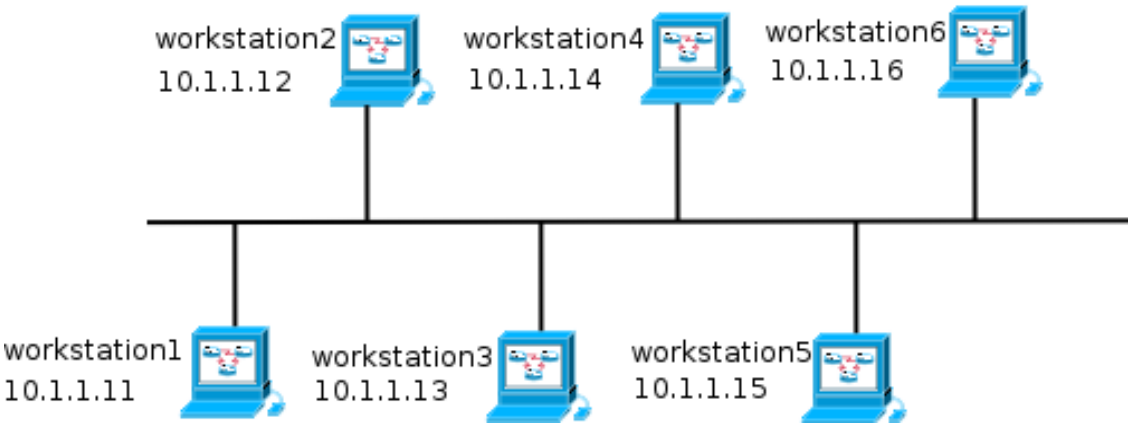

Figura 2.8: Estrutura com o arquivo hosts.txt em cada estação para mapeamento de nomes em endereços IP.

NOME	IP
workstation1	10.1.1.11
workstation2	10.1.1.12
workstation3	10.1.1.13
workstation4	10.1.1.14
workstation5	10.1.1.15
workstation6	10.1.1.16

Figura 2.9: Representação do arquivo hosts.txt em cada estação da rede.

Essa manutenção era feita pela *Network Information Center* (NIC) e, dependendo da quantidade de *hosts* na rede, esse sistema poderia consumir muito da largura de banda disponível, pois era enviado uma nova versão do arquivo para todas as estações, assim que uma atualização era feita. Mas o tempo foi passando e tornando a vida de quem mantinha o sistema de nomes original cada vez mais complicada; o crescimento da rede foi acontecendo e os membros da ARPANET foram se tornando outros sistemas autônomos, com suas próprias estações disponibilizadas em redes locais. Essas, até que administravam seus próprios nomes e endereços, no entanto, continuavam tendo de esperar pela NIC enviar o arquivo HOSTS.TXT para que as alterações fossem visíveis a todos na Internet. Ainda, além das empresas estarem querendo um “lugarzinho” no espaço de nomes, as aplicações foram evoluindo para um patamar bem mais sofisticado e passaram a exigir um sistema de nomes mais eficaz (MOCKAPETRIS, 1987). Obviamente, um crescimento explosivo na quantidade de *hosts* na grande rede tornaria esse sistema insuficiente e impraticável, muito brevemente. Enfatizando isso, Mockapetris (1983) coloca uma citação da própria ARPA Internet:

> A ARPA Internet ilustra a dimensão desse problema: este é um sistema muito abrangente e é provável que cresça muito mais. A necessidade de haver um mapeamento entre nomes de *hosts* (ex., USC-ISIF) e endereços da ARPA Internet (ex., 10.2.0.52) está começando a sublinhar o atual mecanismo. Atualmente os *hosts* na ARPA Internet são registrados pela *Network Information Center* (NIC) e listados numa tabela global (disponível como um arquivo <NETINFO>HOSTS.TXT no *host* SRI-NIC [RFC 810]. O tamanho dessa tabela, e especialmente a freqüência de atualização dessa tabela, está próximo do limite da sua gerenciabilidade. O que se precisa é de uma base de dados distribuída que possibilite a mesma função e, conseqüentemente, evite os problemas causados por uma base centralizada”(ARPA apud MOCKAPETRIS).

Com o sistema de e-mails a dificuldade era ainda maior. Ao acreditarem que a centralização das caixas de e-mails era impossível, os implementadores desse sistema criaram métodos irregulares e gigantes para localizar uma *mailbox*. Alguns desses métodos exigiam que os usuários lembrassem de múltiplas formas de endereçamento, dentre outras técnicas, para enviar uma mensagem, pois os endereços dos e-mails de destino eram compostos por endereços das rotas e dos equipamentos responsáveis pelo encaminhamento dessas mensagens (MOCKAPETRIS).

Esses problemas possuiam características em comum, as quais denotavam algumas sugestões para qualquer solução. Basicamente, precisava-se de um sistema de nomes que pudesse referenciar um recurso sem que houvesse a necessidade de se colocar endereços, rotas, etc. Também, o tamanho da base de dados e suas atualizações, teriam de ser gerenciadas distribuidamente e, além disso, com um cache local para maior agilidade nas consultas. Qualquer sistema computacional usufruiria desse mecanismo; desde um computador pessoal, até um *cluster*, embora pudesse ser de maneiras diferentes. A freqüência das atualizações, como a inserção de novos nomes e endereços, ou deleções desses mesmos dados, bem como o tamanho da base de informações, sugeria um gerenciamento distribuído. Deveria ser de uso generalizado, ou seja, suportar nomes para referenciar *hosts*, caixas de e-mail e outras possíveis informações que viessem a surgir.

Dessa forma, muitas idéias vieram a tona para o estabelecimento de um serviço de nomes mais eficiente e que pudesse ser gerenciado com menos dificuldades. Em meio a tantas sugestões, havia algo em comum, um sistema hierárquico de espaços de nomes separando os níveis pelo caractere “.”. Esse sistema deveria ser constituído por um espaço de nomes, consistente, representando recursos, mas sem nenhum tipo de identificação de redes, endereços, rotas ou quaisquer tipos de dados nesse sentido.

O sistema de DNS que se tornou realidade, transformando-se no sistema de DNS atualmente na Internet, além do mencionado acima, teve muitos outros preceitos para tornar sua concepção uma realidade eficaz, os quais não serão comentados nesse trabalho, pois não é esse o objetivo principal. Todos os detalhes podem ser obtidos na RFC 1034 (1987) (sucessora da RFC 882) e na RFC 1035 (1987) (sucessora da RFC 883).

Para resumir o que é o DNS e como ele é usado, segue um trecho de Tanenbaum (1997, p. 709, 710):

> A essência do DNS é a invenção de um esquema de atribuição de nomes, hierárquico, baseado em domínios. Ele é principalmente usado para mapear nomes de *host* e destinos de mensagens de correio eletrônico em endereços IP, mas também pode ser usado para outros objetivos. O DNS é definido nas RFCs 1034 e 1035.
>
> Para resumir, o DNS é usado da forma descrita a seguir. Para mapear um nome em um endereço IP, um programa aplicativo chama um procedimento de biblioteca denominado **resolvedor** (*resolver*) e passa seu nome para ele como um parâmetro. O resolvedor envia um pacote UDP para um servidor DNS local, que procura o nome e retorna o endereço IP para o resolvedor. Em seguida, o resolvedor retorna o endereço IP para o aplicativo que fez a chamada. Armado com o endereço IP, em seguida o programa pode estabelecer uma conexão TCP com o destino, ou enviar pacotes UDP até ele.

Para exemplificar uma consulta DNS a figura 2.10 ilustra um *host* (10.1.1.2) consultando seu DNS primário (200.175.5.139) pelo

endereço do *site* www.fucap.edu.br. Essa consulta foi executada a partir de um *browser*, para obtenção da página em questão. Nesse caso, a aplicação mencionada por Tanenbaum, no trecho acima, pode ser representada por esse *browser*, que passou o nome para o *resolver* local, que, por sua vez, fez a interação com o servidor DNS configurado no *host* localmente.

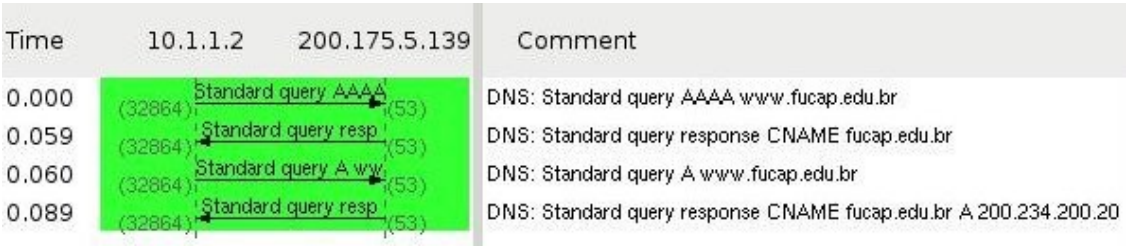

Figura 2.10: Consulta DNS para obter o IP de um servidor *web*.

Nessa consulta DNS nota-se, primeiramente, uma *query* AAAA buscando por um endereço IPv6 de 128 *bits* e, em seguida, a *query* A significando uma busca pelo endereço IPv4 de 32 *bits*.

A figura 2.11 ilustra com mais informações a consulta IPv6 com uma *query*, *type* AAAA e *class* IN; isso quer dizer uma consulta, do tipo IPv6 de Internet.

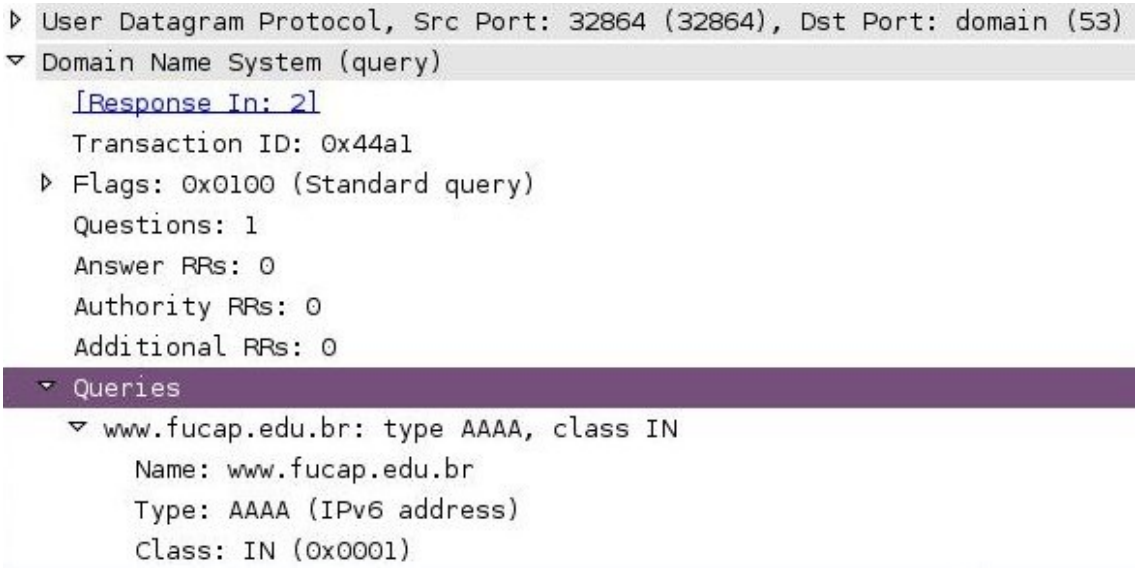

Figura 2.11: Consulta DNS pelo IPv6 do *site* www.fucap.edu.br.

Na figura 2.12 verifica-se através de mais informações uma *query*, *type* A, *class* IN. Isso denota uma consulta, do tipo IPv4 de Internet.

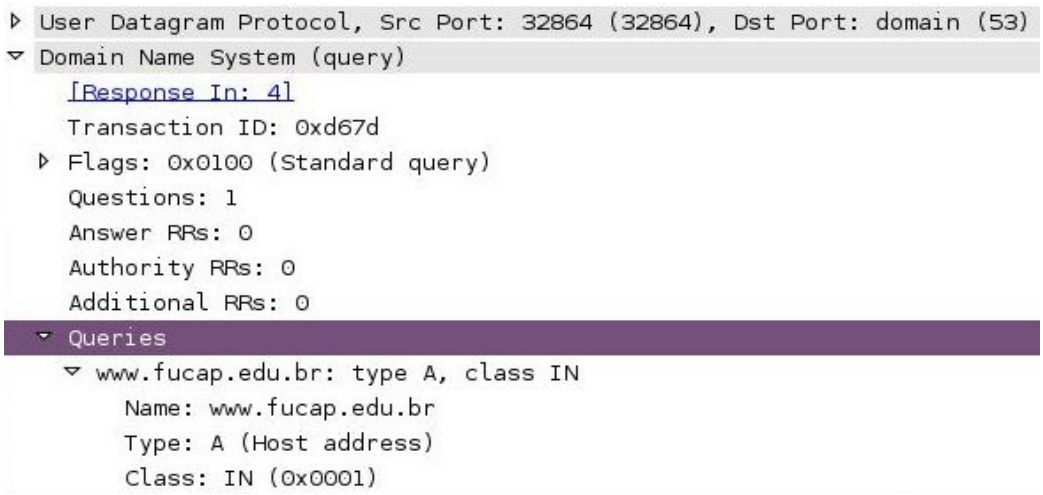

Figura 2.12: Consulta DNS pelo IPv4 do *site* www.fucap.edu.br.

A figura 2.13 identifica a resposta para a consulta IPv6.

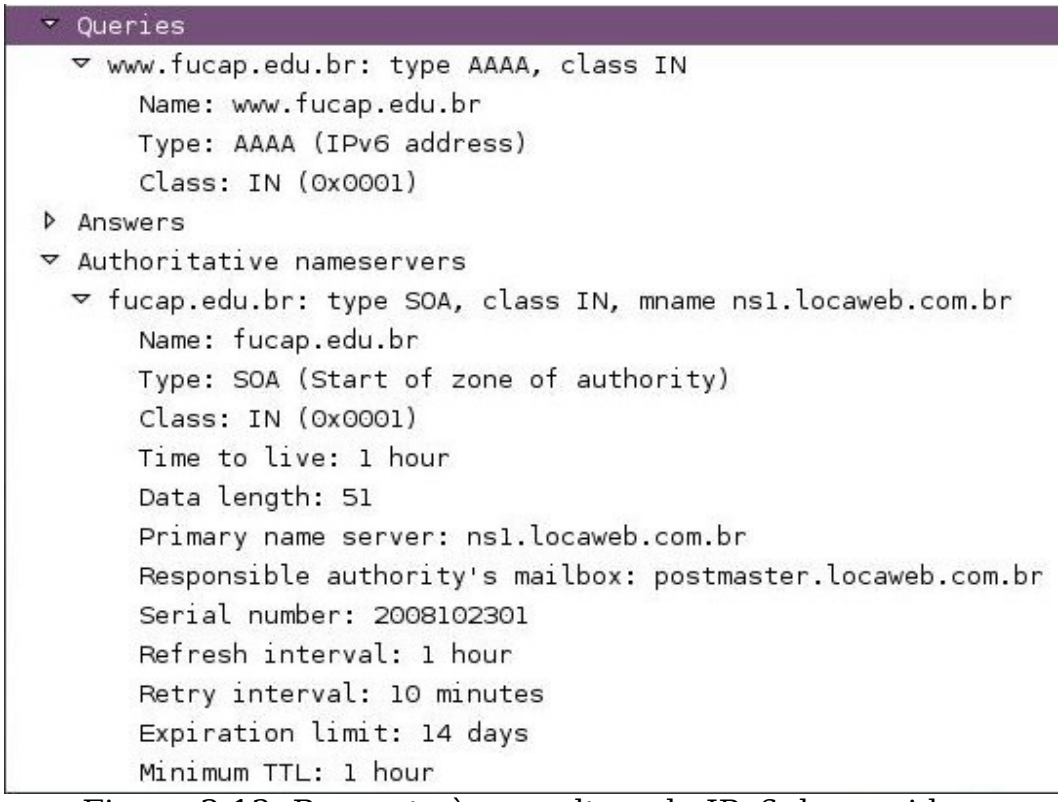

Figura 2.13: Resposta à consulta pelo IPv6 do servidor.

A figura 2.14 exibe a resposta para a consulta IPv4, com o endereço IP efetivamente, dentre várias outras informações.

A partir de então, a conexão TCP para obtenção da página foi possível.

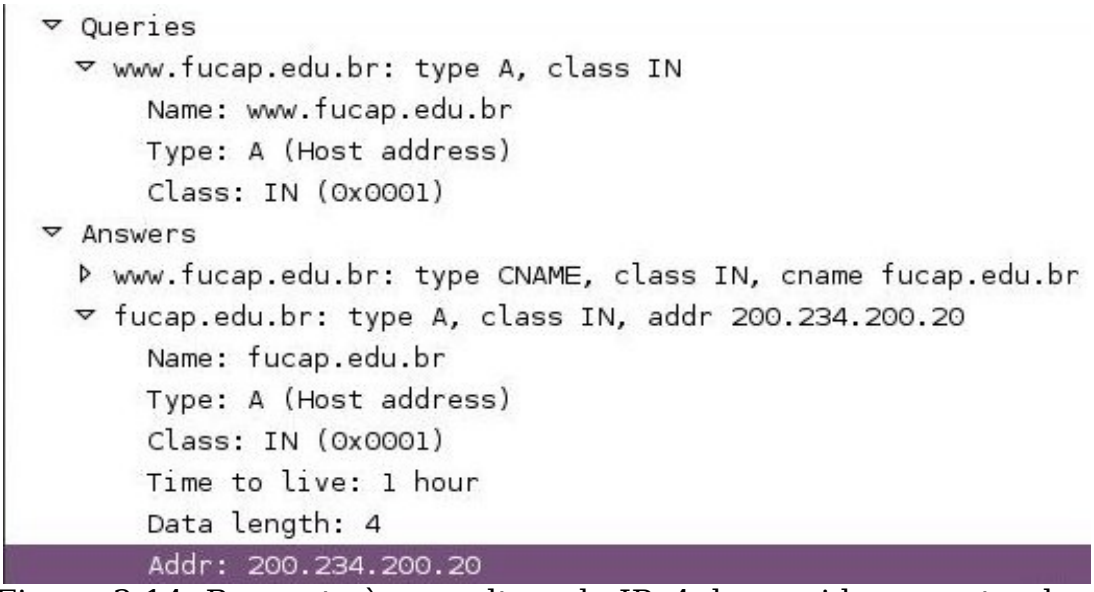

Figura 2.14: Resposta à consulta pelo IPv4 do servidor, constando o IP do servidor, campo *Addr:*.

As figuras demonstradas acima ilustram tanto a consulta pelo IPv6, quanto a consulta pelo IPv4 do servidor www.fucap.edu.br, no entanto, a resposta que realmente trouxe um IP para a efetivação da comunicação, foi a resposta para a *query* IPv4. Isso se deu pelo fato do servidor remoto possuir um IP na versão 4 e não possuir um IP na versão 6. Isso é natural, se for levado em consideração que o IPv4 ainda reina na Internet. Caso ele possuísse um IPv6, a *query* para o IPv6 também seria respondida com o IPv6 válido. Como prova disso, a figura 2.15 ilustra tanto uma consulta por um IPv4, quanto uma consulta pelo IPv6 do site www.ipv6.br, o qual é possuidor desses dois tipos de IPs.

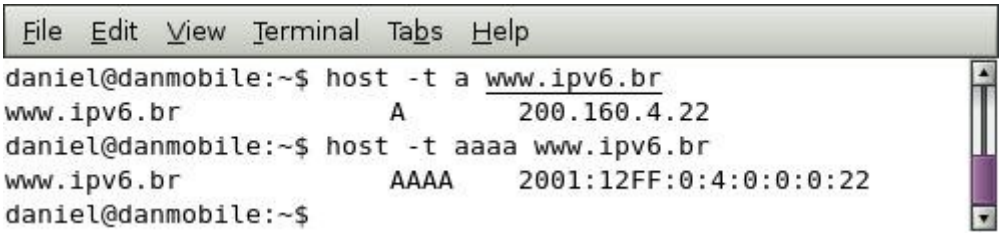

Figura 2.15: Comando *host* consultando o IPv4 e o IPv6 do sítio www.ipv6.br.

Nas consultas realizadas anteriormente, embora tenha havido a participação de apenas um servidor DNS, inserido localmente nas configurações de rede do *host*, o esquema de DNS é muito mais complexo no sentido de haver consistência, agilidade, gerenciabilidade e robustez na estrutura. Geralmente uma estação de trabalho, ou um servidor, têm pelo menos dois servidores de nomes identificados nas configurações de rede. Um desses servidores é conhecido como servidor DNS primário e o outro como servidor DNS secundário. E, dependendo do que se pretende, pode-se ter mais servidores inseridos. Normalmente no servidor primário, considerado o principal, existe um arquivo gravado em disco com

todas as informações de um domínio, ou subdomínio, ou zona, etc. Além do servidor primário, existem os servidores secundários, os quais se alimentam das informações contidas no arquivo localizado no disco do servidor principal.

A quantidade de servidores em uma zona depende da implementação e isso pode variar bastante; a idéia é fornecer um mecanismo de redundância para no caso do servidor primário falhar (ou sobrecarregar) as consultas serão direcionadas para os demais.

Caso o *resolver* receba uma solicitação de qualquer aplicação, ele imediatamente irá consultar o servidor DNS primário configurado no sistema e se o recurso exigido estiver sobre a jurisdição desse servidor DNS então serão retornados registros de recursos oficiais. Um *authoritative record* é fornecido pelo servidor de nomes principal contido no domínio; ou seja, a autoridade que gerencia o registro no domínio e será sempre correto. Por outro lado, os dados em *cache* podem estar desatualizados e, conseqüentemente, equivocados. Os dados em *cache* são controlados pelo campo TTL que é incluído em cada registro de recurso; após expirar o tempo de vida, as informações do *cache* são expurgadas (TANENBAUM). A figura 2.16 denota bem a participação de alguns mecanismos do DNS, em uma configuração típica.

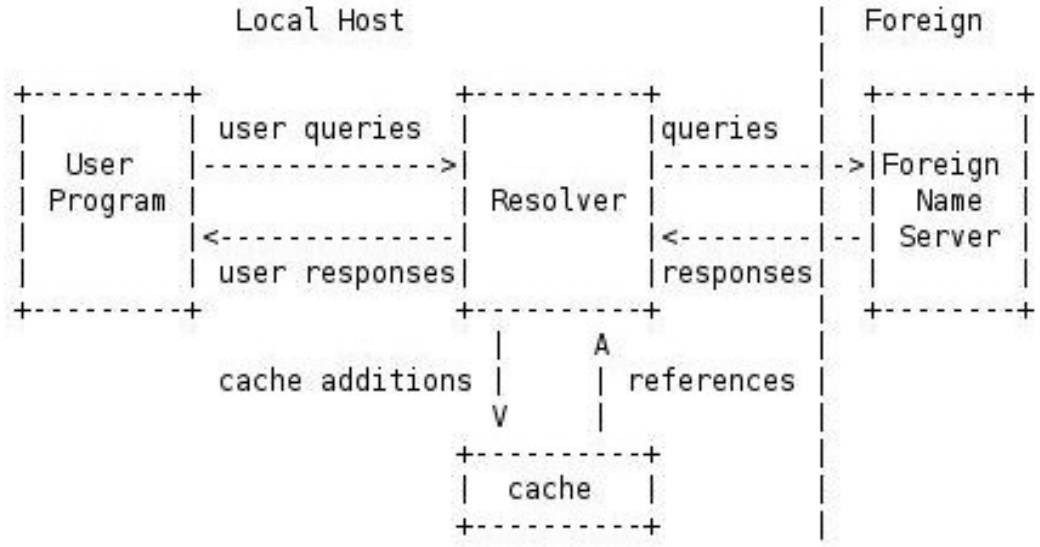

Figura 2.16: Configuração simples do DNS (MOCKAPETRIS,1987).

Se por ventura uma consulta realizada não possa ser resolvida pelo servidor no domínio local, então esse servidor se encarregará de buscar a informação; e isso pode acontecer de uma maneira iterativa ou recursiva.

Segundo a RFC 1034 o DNS tem três grandes elementos, os quais são descritos, concisamente, na tabela 2.1.

Tabela 2.1: Os três maiores componentes no DNS

Componente	Descrição
DOMAIN NAME SPACE e *RESOURCE RECORDS* (RR)	Especificações para o espaço de nomes na estrutura da árvore e os dados associados com os nomes. Conceitualmente, cada nodo e folha da árvore possuem um conjunto de informação, e operações de consultas tentam extrair tipos específicos de informações de um determinado conjunto. Uma consulta de nome para um domínio de interesse, especifica o tipo de informação que é esperada. Por exemplo, a Internet usa alguns desses nomes de domínios para identificar *hosts*; consultas por endereços de recursos, retorna um endereço do *host* na Internet.
NAME SERVERS	São aplicações rodando nos servidores as quais manuseiam as informações sobre a estrutura da árvore do domínio. Um servidor de nomes pode ter uma estrutura ou um conjunto de informações de uma determinada parte de uma árvore de domínio, armazenada em cache, mas, em geral, um servidor de nomes possui informações completas sobre um subconjunto na árvore de um domínio, e vetores para outros servidores de nomes que possam ser usados para conduzir a informação de qualquer parte da árvore. Um servidor de nomes conhece as partes da árvore do domínio das quais ele possui informações completas; um servidor de nomes é chamado de *AUTHORITY* para essas partes do espaço de nomes. A informação autoritativa é organizada em unidades chamadas *ZONEs*, e essas zonas podem ser automaticamente distribuídas para os servidores de nomes os quais provêm serviço redundante para os dados em uma zona. O servidor avalia as informações dessas zonas de tempos em tempos para se manter atualizado. Ele também armazena, em *cache*, outros dados adquiridos pelo resolver local, isso permite mais eficiência com dados que são consultados com alguma freqüência.
RESOLVERS	São aplicações que extraem as informações de um servidor de nomes, em resposta às solicitações dos clientes. O resolvedor deve estar apto a acessar pelo menos um servidor de nomes e usar esse servidor para responder por uma consulta imediatamente, ou prosseguir a consulta usando outros servidores de nomes como referência. Um resolvedor é normalmente uma rotina de sistema que é consultado diretamente pelas aplicações; por isso não é necessário um protocolo entre o programa e o *resolver*.

Fonte: MOCKAPETRIS, 1987.

Vale destacar alguns dos principais tipos de registro de recurso (RR) utilizados pelo sistema de DNS. A tabela 2.2 relaciona isso com uma breve descrição.

Tabela 2.2: Registro de Recursos principais do DNS

A	Endereço IPv4 de um recurso
AAAA	Endereço IPv6 de um recurso
NS	Nome de servidor de domínio ou subdomínio
MX	Servidor de correio eletrônico
CNAME	Nome canônico ou apelido para outro nome do recurso
SOA	Início de autoridade, responsável pelas respostas autoritativas por um domínio
WKS	Descrição de um serviço bem conhecido
PTR	Alias para um endereço IP, isto é, ponteiro para um nome de *host*/domínio reverso a partir de um IP
HINFO	Informações sobre o *host*, como CPU e sistema operacional, por exemplo
TXT	Texto ASCII que serve para descrição; não é interpretado
SRV	Serviços disponíveis em um domínio

Concluindo, o sistema de nomes foi desenvolvido dessa forma para facilitar o gerenciamento, a agilidade e a vida dos usuários e gestores da grande rede, pois permite, principalmente, que os seres humanos possam acessar um recurso na Internet com maior facilidade, ou seja, lembrando do nome do site ao invés de um número IP. Além disso, o gerenciamento distribuído e o uso de caches locais fez do sistema algo bem mais robusto e veloz.

Existem diversos outros parâmetros e funcionalidades, porém, não cabe nesse trabalho um profundo estudo do *Domain Name System*.

2.3.1 DNS Reverso

O DNS reverso é o mecanismo que permite obter o nome de um determinado recurso, através do seu endereço IP. Ou seja, como o nome já diz, é o inverso da tradução de nomes em endereços. Segue um trecho do livro Firewalls e Segurança na Internet de Cheswick, Bellovin e Rubin (2008) para enriquecer o entendimento:

> O espaço de nomes de DNS é estruturado em árvores. Para facilidade de operação, subárvores podem ser delegadas a outros servidores. São utilizadas duas árvores logicamente distintas. A primeira mapeia nomes de *host*, como SMTP.ATT.COM, para endereços como 192.20.225.4. Outras informações por *host* podem ser opcionalmente incluídas, como HINFO ou registros MX. A segunda árvore é para

consultas inversas e contém registros PTR. Nesse caso, ela mapearia 4.225.20.192.in-addr.arpa para smtp.att.com. Não há nenhum relacionamento imposto entre as duas árvores, embora alguns *sites* tenham tentado impor esse *link* para alguns serviços. A árvore inversa raramente é tão bem mantida e atualizada como a árvore de mapeamento direto comumente utilizada.

A figura 2.17 ilustra paralelamente as duas árvores do sistema de DNS.

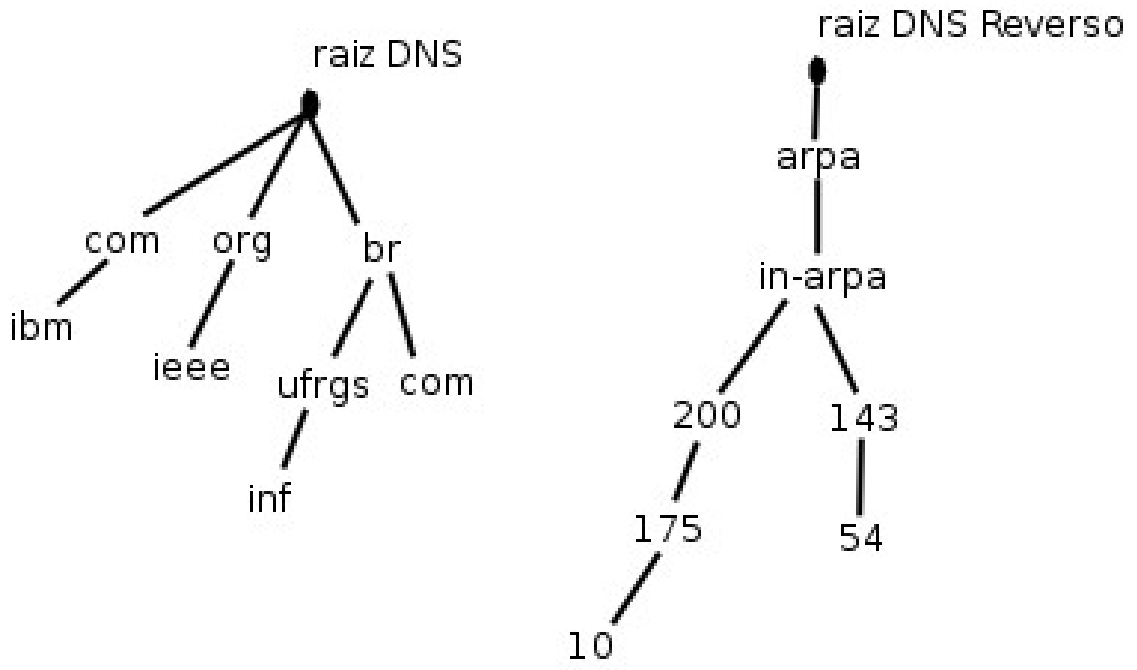

Figura 2.17: Paralelo entre a árvore DNS e a árvore do DNS reverso.

A função do DNS reverso permite a constatação da autenticidade de endereços, fazendo uma consulta no servidor DNS e avaliando se um determinado IP realmente é o correspondente ao domínio. É um mecanismo bastante utilizado para dificultar os spams, pois um *spammer* pode estar forjando um domínio que não lhe pertença para o envio dessas mensagens indesejadas. Como ainda existem muitos servidores de DNS sem as devidas configurações de DNS reverso, conforme já dito anteriormente, há também uma grande incidência de falsos positivos.

3 MÉTODO PROPOSTO

Lembrando o que já foi dito no resumo apenas se propõe um método que possivelmente poderá ser melhorado, e provavelmente reajustado, para funcionar de uma maneira genérica, ou seja, em qualquer sistema de MTA. Um detalhamento mais aprofundado, com a escolha de uma (ou algumas) linguagem de programação, bem como um protótipo funcional será proposto num trabalho futuro.

3.1 Motivação

Conforme mencionado no início desse trabalho, a quantidade de spam é cada vez mais crescente e há estudos na Internet que divulgam algo maior que os cerca de 80%, ditos na introdução, do tráfego de e-mails na grande rede; alguns desses sítios apresentam matérias com algo superior a 90% de spams, dentre todos as mensagens de correio eletrônico. Isso está motivando órgãos, especialistas e universidades a estarem sempre em busca de mecanismos cada vez mais eficientes, no sentido de tentarem diminuir essas estatísticas.

Uma das metodologias empregadas é a aplicação da verificação do DNS reverso, nos domínios que interagem com um MTA local. Porém, nem todos seguem as recomendações do Comitê Gestor da Internet no Brasil (CGI.br):

> 1.4 Serviços DNS Configurados Corretamente
>
> Visando prover identificação imediata dos computadores ligados à Internet brasileira, seguindo um padrão mundial de operação com registros direto e reverso de "Domain Name System" (DNS), faz-se necessário que todas as redes conectadas à Internet/Br implementem tais serviços.
>
> *Recomendação:* Todas as redes conectadas à Internet brasileira devem operar com registros direto e reverso de DNS corretamente configurados (COMITÊ, 2008).

Essa falta de configurações adequadas, do DNS reverso, faz com que ocorram muitos falsos positivos; dessa forma, esse trabalho tem como motivação a verificação dessas configurações em domínios que interajam com o MTA local, e, além disso, permita que o administrador possa, opcionalmente, avisar os administradores remotos dessa falha. Portanto, possibilitando que esse problema possa ser resolvido.

3.2 Método

Primeiramente, algumas premissas:

- sugere-se que a ferramenta funcione num ambiente *web*, ou seja, que possa ser acessado via *browser*; isso irá permitir um funcionamento independente do sistema operacional utilizado;
- a busca das informações de entrada será em arquivo de *log* do MTA;
- quer se encontrar remetentes (IP) válidos os quais interajam com o domínio local e não possuam DNS reverso configurado;
- o objetivo é encontrar postagens válidas; e
- os *spammers*, já detectados através de outros métodos, devem ser ignorados.

A figura 3.1 mostra como poderia funcionar a ferramenta, enquanto que as etapas estão propostas após essa ilustração.

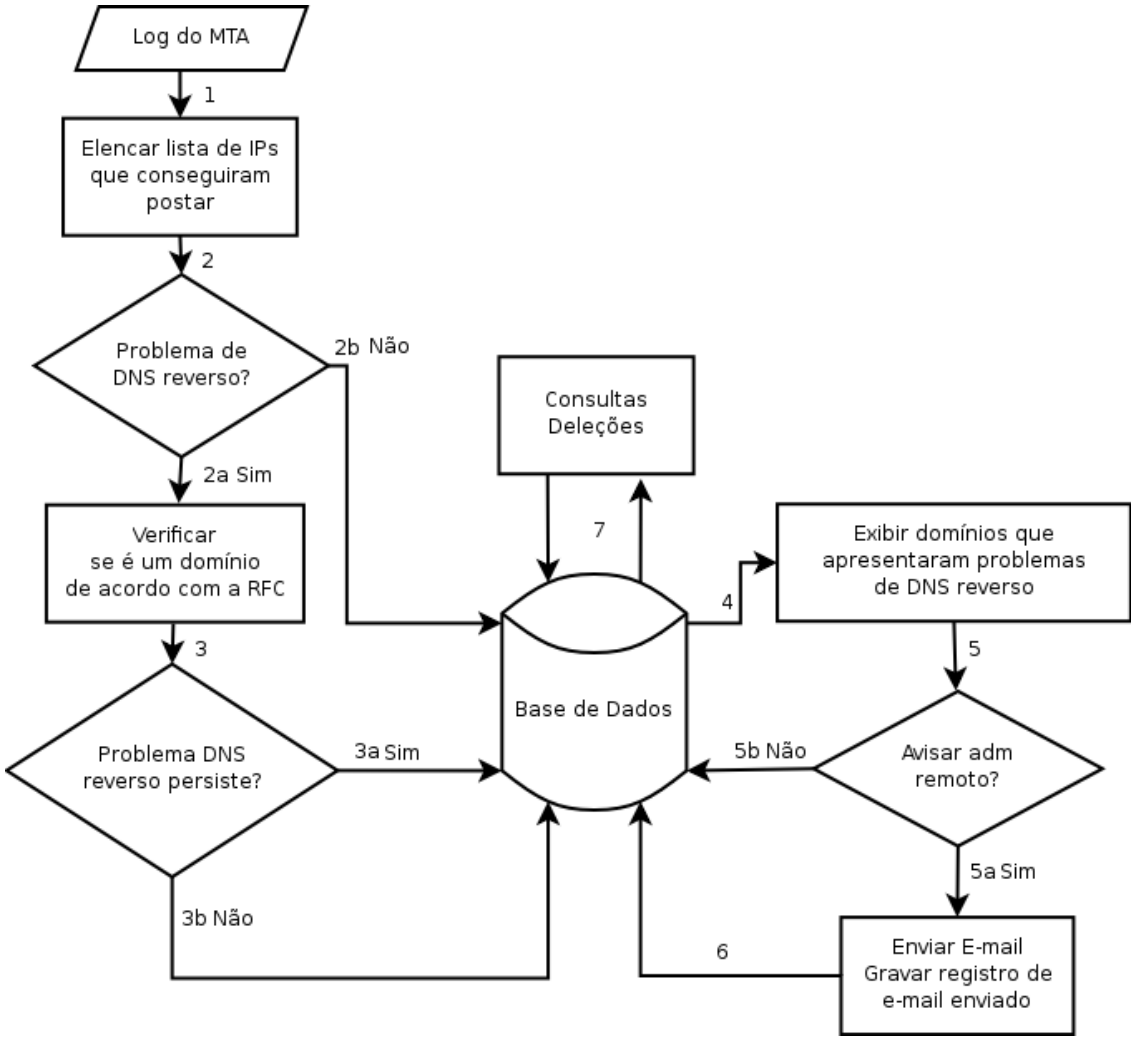

Figura 3.1: Funcionamento sugerido da ferramenta.

Etapas do diagrama disposto na figura 3.1:

Como entrada, o sistema deverá ler as informações contidas no arquivo de log do MTA, armazenado no domínio local. Esse log

poderá estar em arquivo texto ou em uma base de dados. Mas independente da forma, deverá funcionar.

Etapa 1:

Primeiramente, o *log* deve ser lido em busca de IPs que evidentemente postaram mensagens no domínio. Esse endereço deverá ser armazenado. Além desse endereço, também é o caso de já guardar o código da mensagem.

Num segundo momento, deve-se buscar por evidências de autenticação do remetente que postou a mensagem e se o IP usado por ele não faz parte do domínio local. Pois, caso isso seja confirmado, não haverá necessidade de se fazer o teste de DNS reverso. Assim, essas informações de código da mensagem e o IP, respectivamente, podem ser eliminados.

Etapa 2:

Ao chegar nesta etapa tem-se uma lista de IPs remotos que postaram mensagens com sucesso no servidor local, sendo todos candidatos a verificação de DNS reverso.

Etapa 2.a:

Aqui nesse nível, deve-se fazer algumas verificações importantes para tentar, ao máximo, comprovar se quem está enviando mensagens não é um *spammer* e sim um domínio autêntico. Isso pode ser realizado, por exemplo, verificando se o domínio remente pode ser resolvido para RRs MX ou RRs A, ou no caso da consulta retornar uma lista de RRs MX, comparar o IP da postagem com cada IP retornado na consulta do MX. Um indicativo de postagem válida é comparar o IP do servidor de e-mails do remetente e ver se ele faz parte da subrede desse domínio. Klensin (2001, p. 22) diz:

> Somente nomes qualificados de domínios resolvíveis (FQDNs) são permitidos quando nomes de domínio são usados no SMTP. Em outras palavras, nomes que possam ser resolvidos para RRs MX ou RRs A (como discutido na sessão 5) são permitidos, assim como RRs CNAME cujo os alvos possam ser resolvidos, por sua vez, para MX ou RRs A. Apelidos locais ou nomes não qualificados não podem ser usados. [...]

Outra sugestão é fazer uso de consultas SPF, o qual é especificado por Wong (2006), permitindo que um domínio possa explicitamente autorizar os *hosts* que estão habilitados para usar esse nome de domínio, e que o *host* receptor possa estar verificando essa autorização. Dessa maneira, podem ser atribuídas pontuações as quais no final poderiam classificar um domínio como legítimo ou não. A figura 3.2 denota um subsistema para classificação dos domínios.

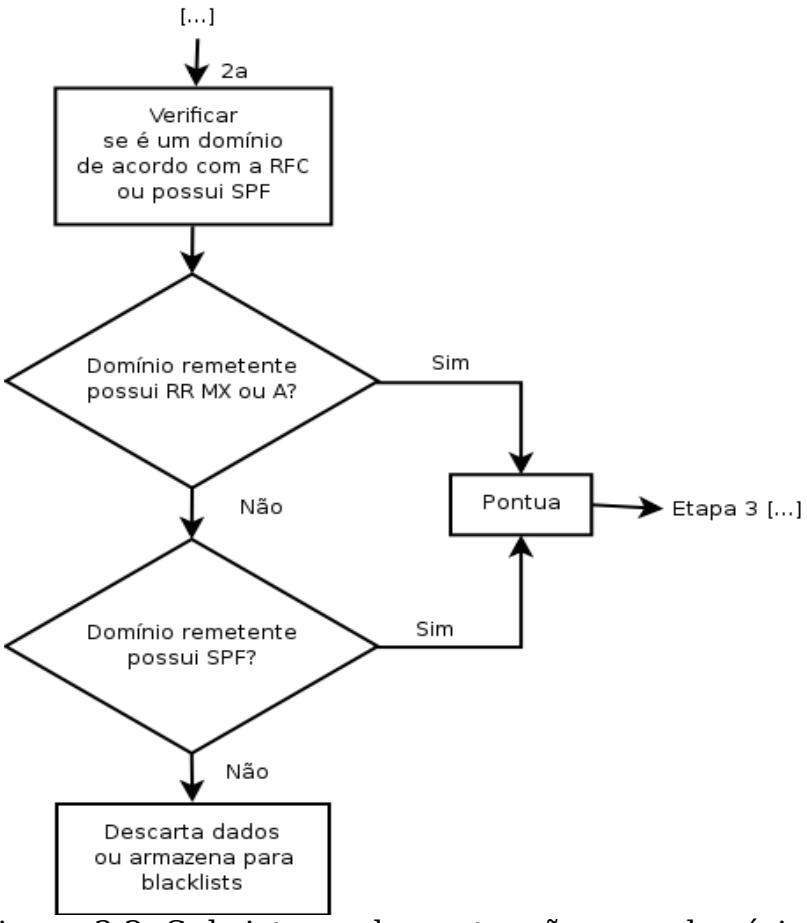

Figura 3.2: Subsistema de pontuação para domínios.

Se com as constatações realizadas for identificado que o domínio possua evidências de ser legítimo, então, aplica-se o teste de DNS reverso. Caso contrário, as informações poderão ser descartadas ou, ainda, serem gravadas para a criação de *blacklists*, por exemplo.

Etapa 2.b:

No caso do DNS reverso estar bem configurado no domínio remoto, pegar esse domínio e armazenar na base de dados, para futuras análises ou simples cadastro.

Etapa 3:

É importante salientar que durante um teste de DNS reverso, assim como qualquer outra comunicação via Internet, podem ocorrer problemas inesperados. O domínio remoto pode estar em manutenção, ou pode-se haver perda de pacotes por motivos de inconsistência ou indisponibilidade de recursos, dentre outras dificuldades. Além disso, um teste pode estar sendo realizado baseado em informações de *logs* um pouco mais antigos e o domínio envolvido já poderia estar configurado adequadamente. Dessa forma, essa etapa possibilita fazer uma nova verificação, na tentativa de garantir a má configuração do DNS reverso.

Etapa 3a:

Se durante um novo teste for realmente constatado que o DNS reverso do domínio remoto esteja com problemas, esse domínio deve ser armazenado numa base de dados.

Etapa 3b:

Nesse nível, constata-se que o domínio remoto possui o DNS reverso configurado sem problemas. Assim, pode-se armazenar os dados para futuras avaliações, como estatísticas de envio de e-mails, ou para inserção em listas brancas, etc. Esses dados podem ser incrementados, pois na etapa 2 eles já poderiam terem sido gravados.

Etapa 4:

Nessa etapa exibe-se uma lista de domínios que apresentaram problema de DNS reverso. Podendo serem relacionados por quantidade de ocorrência. Além disso, podem ser mostrados o número de avisos já emitidos por e-mail. Isso ajudará ao administrador tomar algumas decisões como, por exemplo, não avisar mais o administrador remoto; avisar automaticamente, sem perguntar; bloquear o domínio definitivamente, etc.

Etapa 5:

Nesse momento é possível que o administrador avise ou não os administradores dos outros domínios. Por motivos diversos, pode acontecer de não ser possível enviar o e-mail de aviso, por exemplo.

Esse e-mail será enviado para o postmaster@dominio conforme determina a RFC 2142.

Etapa 5a:

Caso o administrador escolha enviar o e-mail, avisando do problema de DNS reverso, isso poderia ser feito de algumas formas como, abrindo o MUA do administrador local ou enviando via rotina do próprio sistema, etc.

O envio dessa mensagem poderia ser automatizado, ou seja, um mecanismo que envie o e-mail sem a intervenção do administrador. No entanto, isso poderia gerar algum desconforto.

Etapa 5b:

Nessa etapa pode-se optar por não avisar momentaneamente. Ou, ainda, escolher não avisar mais pelo fato de já ter sido avisado por diversas outras vezes, ou por uma quantidade pré-definida. Outro motivo seria o domínio remoto já ter respondido que não fará a inserção de dados correspondentes ao DNS reverso, por um motivo qualquer.

Etapa 6:

É o instante em que efetivamente será enviado o e-mail de aviso e, por conseguinte, deve ser gerado um registro de emissão do alerta, na base de dados, ao respectivo domínio. Caso já tenha sido enviado outras vezes, isso deve ser um incremento.

Etapa 7:

A etapa 7 poderia ser como um *front-end* mais apurado, isto é, uma *interface* através da qual o administrador poderia estar gerando diversos relatórios, no sentido de estar verificando quais domínios interagem com o MTA local, corretamente ou não; quais domínios foram avisados e quantas vezes o aviso já foi emitido; pode-se estar fazendo manutenções que envolvam deleções de informações desatualizadas, como e-mails que não funcionam; domínios que deixaram de existir, etc.

3.3 Validação do Método

A validação do método proposto pretende ser através da averiguação de arquivos de *logs* reais, com números aproximados. Para isso, foram utilizados os *logs* gerados pelo MTA do Instituto de Informática da UFRGS. Nesse caso, o *software* de MTA é o Postfix [POSTFIX, 2008], do qual não se pretende fazer nenhuma menção tecnicamente aprofundado.

Esses *logs* foram concebidos no período do mês de setembro de 2008, em um arquivo por dia, os quais somados tiveram cerca de 1.2 GB de dados. A análise foi baseada em todos os arquivos, demonstrando-se, dessa forma, informações sobre os e-mails postados por remetentes que não fossem autenticados no servidor e que não fizessem parte do domínio inf.ufrgs.br. Ou seja, os dados que interessaram foram os de remetentes externos postando para usuários com *mailbox* no domínio da Universidade Federal do Rio Grande do Sul.

A etapa 1 da proposta é o levantamento de informações que comprovem uma postagem legítima. Nos dados do MTA, mencionados acima, a tarefa foi encontrar indicações que comprovassem uma postagem, conforme segue:

- buscou-se por indícios de postagem, verificando toda linha que contivesse um *queue_id*, de mensagem, com seu respectivo remetente. Mas que não fosse de alguém no domínio local. Exemplos:
  - 5FAAF61CB4: *client=unknown*[207.XX.XXX.XX].
  - C003F61C91: *client*=yx-out-2324.google.com[74.125.44.28].
- Além disso, foi preciso saber se essa mensagem não houvesse sido descartada. Exemplo:
  - 5FAAF61CB4: *reject*.

Através dos passos acima, foram confirmados 121.566 postagens, de domínios externos, totalizando 119.727 mensagens postadas efetivamente.

Na etapa 2 está mencionado que deve-se testar os DNSs reversos nos IPs relacionados na etapa 1, os quais somaram 10.128 diferentes IPs. Novamente isso foi realizado através da linha de comando no Linux, demonstrando que 8604 IPs estavam com seus DNSs reversos

configurados adequadamente, enquanto que os demais apresentaram as mensagens abaixo, conforme saídas do comando `host -t ptr ip_destino`:

- 1390 – *does not exist, try again*;
- 124 – PTR *record not found, server failure*;
- 7 – *Nameserver not responding, PTR record not found, try again*;
- 2 – PTR *record not found*; e
- 1 – PTR *record currently not present.*

A etapa 2a é para tentar garantir que domínios não autênticos, como por exemplo os de *spammers*, não sejam avisados do problema. Dessa forma, após realizados os primeiros testes de DNS reverso, é o momento de verificar se os remetentes, com esses problemas, realmente têm seus domínios configurados de acordo com as recomendações da RFC; assim, um dos testes possíveis é o de verificação de um MX configurado. Essa averiguação pode ser feita com a utilização do comando *whois* em conjunto com o comando *host*. Dessa forma, poderia-se constatar, também, se o IP do remetente está contido na sub-rede do domínio avaliado ou se ele é um RR MX desse domínio. Exemplo:

Comando *whois*:

whois 143.107.106.107

A saída foi suprimida, para avaliar as informações que interessam. Aqui pode-se notar que, apesar da verificação do DNS reverso no IP 143.107.106.107 ter retornado erro, trata-se de um IP que está dentro de uma sub-rede do domínio usp.br:

```
inetnum:      143.107/16
nserver:      BEE.USPNET.USP.BR
```

Com isso, pode-se fazer a verificação das configurações de MX nesse domínio, conforme abaixo:

Comando *host*:

```
host -t mx usp.br
```

Saída:

```
usp.br                  MX      0 kavir.uspnet.usp.br
usp.br                  MX      0 sinai.uspnet.usp.br
```

Nesse ponto, pode-se notar que existem RRs MX configurados no domínio e, agora, nota-se que o IP 143.107.106.107 não é nem um dos RRs MX retornados, conforme comandos abaixo:

```
host -t a kavir.uspnet.usp.br
kavir.uspnet.usp.br     A       143.107.254.93
host -t a sinai.uspnet.usp.br
```

```
sinai.uspnet.usp.br     A       143.107.254.76
```

Finalmente, verifica-se a configuração adequada do DNS reverso nos RRs MX retornados:

```
host -t ptr 143.107.254.93

Name: kavir.uspnet.usp.br

Address: 143.107.254.93

host -t ptr 143.107.254.76

Name: sinai.uspnet.usp.br

Address: 143.107.254.76
```

Com as informações de saída dos comandos acima, verificou-se que o IP 143.107.106.107, no qual foi feito teste de DNS reverso e retornou erro, faz parte de uma rede com configurações adequadas de MX e DNS reverso. Não precisando, dessa forma, avisar o administrador dessa rede.

Possivelmente o IP em questão possa ter sido um MX do domínio válido em setembro e que no momento dos testes, ou seja, cerca de dois meses depois dos *logs* serem gerados, esse IP tenha sido trocado pelos IPs dos RRs MX atuais. Portanto, não há necessidade de avisar sobre o problema de DNS reverso.

Além das consultas de MX é possível serem executados outros testes os quais avaliariam as configurações de SPF, nos IPs que apresentaram problema de DNS reverso. Lembrando que o motivo maior desses testes é permitir ao administrador local, a escolha de avisar sobre a má configuração do DNS reverso aos domínios considerados legítimos. Para a avaliação do funcionamento de SPF no domínio remoto, conforme Michellis (2005) sugere em seu *site*, é possível utilizar o comando `host -t txt dominio.com.br`. Segundo Michellis “Se quiser saber qual o registro SPF de um domínio, basta rodar essa consulta[...]”.

Além disso, vale mencionar o que o Antispam.br (2008) diz sobre a publicação da política SPF:

> Ao publicar uma política de SPF, o administrador de um domínio está autorizando determinados MTAs a enviar *e-mails* em nome deste domínio.[...]
>
> Estas políticas são publicadas através de registros TXT do DNS, em formato ASCII. Um exemplo desse registro é:
>
> **Exemplo**:
>
> example.com. IN TXT "v=spf1 a mx ip4:192.0.2.32/27 -all"
>
> Neste caso a política estabelece que pode enviar mensagens em nome do domínio example.com uma máquina que satisfaça um dos seguintes critérios:
>
> - seu endereço IP deve ser um RR tipo A do domínio example.com (a);
> - seja designada como MX do domínio example.com (mx); ou

- pertença ao bloco de endereços IP 192.0.2.32/27 (ip4).

> A cláusula "-all" diz que devem ser recusados ("-", prefixo *Fail*) *e-mails* partindo de qualquer outro endereço IP (all).[...]

Com os resultados acima, já seria possível pontuar os IPs que retornaram respostas bem sucedidas dos RRs MX e, a partir daí, o administrador local já poderia estar tomando alguma decisão em relação ao aviso de DNS reverso mal configurado, do domínio remoto. Porém, a proposta recomenda que outro teste seja feito, pois conforme já dito, os primeiros testes de DNS reverso poderiam terem retornado erros em momentos de inconsistências, manutenções, averiguação de *logs* antigos, ou qualquer outro motivo que pudesse dificultar essa consulta.

Sendo assim, na etapa 3, pretende-se um novo (ou mais) teste, de modo que na etapa 3a, caso o problema de DNS reverso persista, ele deve ser armazenado na base de dados.

A partir da etapa 2b, até a etapa 7, finalmente, a proposta sugere um mecanismo para armazenamento, exclusão, análises estatísticas e propriamente a opção de avisar ou não o remetente de uma e-mail, o qual faça parte de um domínio com má configuração ou inexistência de DNS reverso. Sendo assim, há diversas formas disso poder ser aplicado e o limite estará na capacidade do desenvolvedor.

4 CONCLUSÃO

A quantidade de spam trafegando na grande rede é realmente impressionante. Embora possa ser bastante reduzida caso haja a persistência, a criatividade e a inovação; tanto por intermédio de órgãos, especialistas e universidades, quanto por parte dos usuários. As instituições investindo com treinamentos, profissionais capacitados, pesquisa e desenvolvimento, e os usuários sendo mais zelosos, mais críticos, mais conscientes e criteriosos.

De fato, por intermédio dos experimentos acima, comprovou-se que existem muitas postagens de domínios com suas configurações deficientes de DNSs reversos. Portanto, viu-se que dos 10.128 IPs, responsáveis por postarem 119.727 mensagens no MTA do Instituto de Informática da UFRGS, no mês de setembro de 2008, 84,95% apresentaram respostas de DNS reverso bem sucedidas; em contra partida, 15,05% desses IPs demonstraram algum problema no momento dessa mesma consulta.

O método proposto, embora possa precisar de ajustes e o desenvolvimento de um protótipo funcional, poderá ser bastante promissor, auxiliando os diversos administradores de redes a buscarem configurar adequadamente os servidores DNSs os quais, por ventura, estejam administrando. Com isso, o mecanismo de DNS reverso aos poucos irá se transformando, cada vez mais, funcional e mais eficiente na batalha contra os spams.

REFERÊNCIAS

ANTISPAM.BR. **História**: origem e curiosidades. Disponível em: <http://www.antispam.br/historia/>. Acesso em: nov. 2008.

ANTISPAM.BR. **Como gerenciar filtros anti-spam**. Disponível em: <http://www.antispam.br/prevencao/filtros/>. Acesso em: nov. 2008.

ANTISPAM.BR. **SPF**: Sender Policy Framework. 2008. Disponível em: <http://www.antispam.br/admin/spf/>. Acesso em: nov. 2008.

AZNAR, G. **The Linux Electronic Mail Administrator HOWTO**. [S. l.], 2000. Disponível em: <http://www.tldp.org/HOWTO/Mail-Administrator-HOWTO-3.html#ss3.1>. Acesso em: ago. 2008.

BINMAIL. 1998. Disponível em: <http://pdg.uow.edu.au/binmail/index.html>. Acesso em: nov. 2008.

CERT.BR. **Cartilha de Segurança para a Internet Versão 3.1**. 2006. <http://cartilha.cert.br/spam/sec1.html#sec1>. Acesso em: nov. 2008.

CHESWICK, W. R.; BELLOVIN, S. M.; RUBIN, A. D. **Firewalls e Segurança na Internet**: Repelindo o hacker ardiloso. 2.ed. Porto Alegre: Bookman, 2005.

COMITÊ GESTOR INTERNET BRASIL. **Recomendações para o Desenvolvimento e Operação da Internet no Brasil**. Disponível em: <http://www.cgi.br/publicacoes/documentacao/desenvolvimento.htm > . Acesso em: nov. 2008.

CONE: Console Newsreader And Emailer. Disponível em: <http://www.courier-mta.org/cone/index.html>. Acesso em: nov. 2008.

CROCKER, D. **Mailbox Names For Common Services, Roles And Functions**: RFC 2142. [S.l.], 1997. Disponível em: <http://www.ietf.org/rfc/rfc2142.txt>. Acesso em: dez. 2008.

EVOLUTION. Disponível em: <http://projects.gnome.org/evolution/>. Acesso em: nov. 2008.

GUDKOVA, D.; MASLENNIKOV, D. **Spam Evolution: April – June 2008**. [S.l.], 2008. Disponível em: <http://www.viruslist.com/analysis?pubid=204792013>. Acesso em: dez. 2008.

KLENSIN, J. **Simple Mail Transfer Protocol**: RFC 2821. [S.l.], 2001. Disponível em: <http://www.ietf.org/rfc/rfc2821.txt>. Acesso em: nov. 2008.

MAIL. **GNU Mailutils Manual**. Disponíel em: <http://www.gnu.org/software/mailutils/manual/mailutils.html>. Acesso em: nov. 2008.

MAIL Transfer Agent. Disponível em: <http://en.wikipedia.org/wiki/Mail_transfer_agent>. Acesso em: nov. 2008.

MARSHAL. **Sex, Drugs and Software Lead Spam Purchase Growth**. August 19, 2008. Disponível em: <http://www.marshal.com/pages/newsitem.asp?article=748&thesection=news>. Acesso em: set. 2008.

MICHELLIS, D. **Adicionando Verificação de SPF ao Postfix**. 2005. Disponível em: <http://www.unitednerds.org/thefallen/docs/index.php?area=Postfix&tuto=SPF>. Acesso em: nov. 2008.

MICROSOFT. **Microsoft Office Outlook**. Disponível em: <http://office.microsoft.com/en-us/outlook/default.aspx>. Acesso em: nov. 2008.

MOCKAPETRIS, P. **Domain Names – Concepts And Facilities**: RFC 1034. [S.l.], 1987. Disponível em: <http://www.ietf.org/rfc/rfc1034.txt>. Acesso em: ago. de 2008.

MOCKAPETRIS, P. **Domain Names – Concepts And Facilities**: RFC 882. [S.l.], 1983. Disponível em: <http://www.ietf.org/rfc/rfc882.txt>. Acesso em: out. de 2008.

MOCKAPETRIS, P. **Domain Names – Implementation And Specification**.: RFC 1035. [S.l.], 1987. Disponível em: <http://www.ietf.org/rfc/rfc1035.txt>. Acesso em: nov. 2008.

MUTT. **The Mutt E-Mail Client**. Disponível em: <http://www.mutt.org/>. Acesso em: nov. 2008.

POSTFIX. **The Postfix Home Page**. Disponível em: <http://www.postfix.org/>. Acesso em: nov. 2008.

SOPHOS REVEALS “DIRTY DOZEN” SPAM-RELAYING COUNTRIES FOR Q3 2007. [S.l.], 2007. Disponível em: <http://www.sophos.com/pressoffice/news/articles/2007/10/dirtydozoct07.html>. Acesso em: nov. 2008.

SUN, C. **Spam filters**: Making them work. September 22, 2008 (Computerworld). p. 1. Disponível em: <http://computerworld.com/action/article.do?command=viewArticleBasic&taxonomyName=security&articleId=325388&taxonomyId=17&intsrc=kc_feat>. Acesso em: set. 2008.

SYLPHEED. **Lightweight and User-friendly E-mail Client**. Disponível em: <http://sylpheed.sraoss.jp/en/>. Acesso em: nov. 2008.

SYMANTEC MESSAGING AND WEB SECURITY. **The State of Spam – A Monthly Report – September 2008**. p. 4. Disponível em: <http://eval.symantec.com/mktginfo/enterprise/other_resources/b-state_of_spam_report_08-2008.en-us.pdf>. Acesso em: set. 2008.

SYMANTEC MESSAGING AND WEB SECURITY. **The State of Spam – A Monthly Report – September 2007**. p. 1. Disponível em: <http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/SpamReport_September07.pdf>. Acesso em: set. 2008.

TANENBAUM, A. S. **Redes de Computadores**. Rio de Janeiro: Campus, 1997. p. 737-757.

THUNDERBIRD 2. Disponível em: <http://br.mozdev.org/thunderbird/>. Acesso em: nov. 2008.

WONG, M.; SCHLITT, W. **Sender Policy Framework (SPF) for Authorizing Use of Domains in E-mail, Version 1**. RFC 4408. [S.l.], 2006. Disponível em: <http://www.ietf.org/rfc/rfc4408.txt>. Acesso em: nov. 2008.

ZOMBIE Lab Online Statistics. <http://www.commtouch.com/site/Resources/ZombieMonitor.asp>. Acesso em: nov. 2008.