PULL: Um novo modelo para o Correio Eletrônico

Antonio Caminada, Luiz Magalhães, Michael Stanton

***Resumo*-Este artigo apresenta um modelo de transporte de email que divide a transferência em duas partes: o cabeçalho é enviado usando SMTP, mas a mensagem é armazenada em um novo servidor, proposto neste artigo, e pode ser acessado via HTTP. Este modelo tem várias vantagens sobre o modelo usado atualmente, a principal sendo requerer um servidor com presença constante na Internet para permitir a busca de email, o que pode inibir o SPAM.**

***Palavras-Chave* – Correio Eletrônico, SMTP, HTTP, MIME**

***Abstract*-This paper presents a new model for sending electronic mail, which breaks the transfer in two parts: the email header is sent using SMTP, but the body is kept in a new server, defined in this paper, and can be downloaded via HTTP. This model has several advantages over the current model, the main advantage being the requirement of a server that has to be connected full time to the Internet to allow the recipients of email to have access to it, which may inhibit SPAM.**

***Keywords*-email, SMTP, HTTP, MIME**

I. INTRODUÇÃO

O email tornou-se uma ferramenta indispensável na vida moderna, seja no trabalho ou no lazer. Com ele podemos enviar mensagens de texto e arquivos (imagens, vídeos, planilhas e etc) rapidamente e a um custo muito baixo. Essas facilidades, aliadas à fragilidade de alguns programas "front end" de email, acabaram se tornando uma arma na mão de pessoas mal intencionadas, que utilizam o email para difundir vírus, *worms* e SPAM. Este trabalho tem por objetivo propor uma alternativa ao atual modelo de correio eletrônico, para minimizar estes e outros problemas.

A idéia do presente trabalho é substituir o modelo "*Push*" utilizado hoje por um modelo "*Pull*", isto é, em vez das mensagens serem automaticamente enviadas para o servidor do usuário, a mensagem ficaria em um novo servidor (proposto neste artigo) e o usuário receberia apenas o cabeçalho da mensagem. À sua conveniência, o usuário poderia escolher quais mensagens trazer para seu computador. Isso minimiza os problemas de vírus e SPAM indiretamente, através da identificação das contas (ou servidores) responsáveis pela disseminação destes emails indesejados, e coloca o ônus do armazenamento da mensagem no distribuidor, reduzindo o tráfego nos servidores de email da Internet (pois mensagens só são transferidas sob demanda, e mensagems indesejadas não ocupam espaço na caixa de correio do usuário).

Na Seção II será feita uma breve exposição sobre o correio eletrônico, contendo um rápido histórico, e seu funcionamento na Internet. Na Seção III falaremos dos principais problemas que afetam o email hoje: SPAM, vírus e o volume de tráfego gerado na Internet e o que está sendo feito atualmente para combatê-los. Na Seção IV, detalharemos nossa proposta de mudança, mostrando as vantagens do novo modelo. Por fim indicaremos os próximos passos da pesquisa.

Antonio Caminada (antonio@dyn.com.br) e Luiz Claudio Schara Magalhães (schara@telecom.uff.br), Departamento de Engenharia de Telecomunicações, Escola de Engenharia;
Michael Stanton (michael@ic.uff.br), Instituto de Computação;
Universidade Federal Fluminense, Niteroi, Brasil.

II. O EMAIL

O email é uma mensagem de texto, enviada de um usuário para outro através da Internet (ou de uma rede local). Mesmo quando contém imagens, sons, arquivos anexados, estes são convertidos em texto através do MIME (Multipurpose Internet Mail Extensions[6]).

Analogamente ao correio tradicional, uma mensagem de correio eletrônico tem duas partes: a mensagem propriamente dita e um "envelope" (cabeçalho), que contém informações de controle para entrega (por exemplo, o endereço do remetente e do destinatário). Assim, a mensagem é formada por um cabeçalho com uma estrutura bem definida e um corpo (texto), que pode ser estruturado (usando MIME).

O formato de email atual surgiu em 1971 pelas mãos de Ray Tomlinsom [1], que na época trabalhava em um programa que permitiria que usuários compartilhando um mesmo computador deixassem mensagens uns para os outros. Ao mesmo tempo ele também desenvolvia um programa para transferência de arquivos entre máquinas ligadas à ARPANET. Tomlinson percebeu que, se juntasse os dois programas, poderia enviar mensagens através da rede.

Um de seus problemas era como distinguir as mensagens que eram para a máquina em uso e para fora (rede). Ele teve a idéia então, de utilizar o símbolo @ (at). Segundo Tomlinson: "Ele designa um lugar e é a única preposição do teclado". [1]

O email da Internet também descende dos programas de BBS, como a FIDONET [12,13,14] e a Usenet[15] e do correio eletrônico da BITNET [16]. Destes ele herdou o modelo de *push*. Tanto a FIDONET quanto a BITNET eram redes do tipo *store-and-forward*. Para a carta chegar ao destino, ela passava por vários sistemas intermediários. Enquanto na BITNET o endereço era plano (usuário@sistema) , na FIDONET o endereço era hierárquico (como na Internet, mas com zona:rede/nó.ponto), e na Usenet o endereço continha as informações de roteamento explícito da carta, sendo uma sequência de nomes de máquina separados por pontos de exclamação (*source-routing*). A conectividade dos sistemas de BBS e do UUCP era esporádica. A entrega de mail dependia dos sistemas intermediários entrarem em contato, na maior parte das vezes via linhas discadas. Por isto, fazia sentido levar a carta para mais perto do usuário (usando o modelo *push*). A cada contato, cartas fluiam de um sistema para o outro. Este contato podia ocorrer a cada hora ou até mesmo uma vez por dia (ou menos), dependendo do custo (ligações de longa

distância, como conexões internacionais, ocorriam com baixa freqüência). Apesar da Internet não requerer o modelo *push*, já que os sistemas normalmente ficam constantemente conectados, antes da Internet se tornar ubíqua era bem entendido que o valor do correio eletrônico aumentava conforme fosse maior o número de pessoas conectadas a ele. Assim, era importante que o correio da Internet fosse compatível com o modelo existente até então (e foram criados vários gateways para permitir o fluxo de mensagens para sistemas de correio diferentes). Daí o modelo *push* ter sido adotado, e funcionar até hoje, apesar de ser o único serviço da Internet que segue este modelo (os outros, como webcasting [17], não obtiveram o mesmo sucesso e são pouco comuns atualmente). Maiores informações sobre a história do correio eletrônico podem ser encontradas em [18].

Na Figura 1 podemos ver os protocolos usados para o transporte de email na Internet. Atualmente o correio eletrônico funciona da seguinte forma: o usuário escreve a mensagem em um "user agent" ou UA, que é um programa utilizado para compor, enviar e receber mensagens, como por exemplo o outlook express. Ele esconde do usuário os detalhes de implementação do sistema. Ao clicar em "enviar" a mensagem é remetida ao servidor SMTP do cliente. O SMTP (Simple Mail Transfer Protocol) especificado na RFC 821 [3] é o protocolo encarregado de encaminhar as mensagens de email do remetente ao destinatário. Ele se baseia no seguinte modelo de comunicação: como resultado de um comando do usuário, o SMTP remetente estabelece um canal bi-direcional com o SMTP destinatário, que pode ser o destinatário final ou um intermediário. Os comandos SMTP são gerados pelo SMTP transmissor e enviados para o SMTP receptor. As respostas são geradas pelo receptor em conseqüência aos comandos. Uma vez estabelecido um canal de comunicação, o transmissor SMTP envia um comando MAIL, indicando o remetente da mensagem. Se o receptor SMTP puder aceitar a mensagem ele responde com um OK. O transmissor, então, envia um comando RCPT, indicando o destinatário da mensagem. Se o receptor SMTP puder aceitar mensagens para o destinatário ele responde com um OK. Caso contrário responde com um comando rejeitando aquele usuário (mas não acaba com a transação). O transmissor e o receptor podem negociar vários destinatários. Quando todos os destinatários tiverem sido negociados o transmissor envia um comando DATA. O receptor trata tudo após esse comando como o corpo da mensagem, que termina com uma linha com um ponto "." apenas. Durante o caminho entre o remetente e o destinatário final a mensagem pode passar por vários MTAs - Mail Transfer Agents – agentes intermediários entre o remetente e o destinatário da mensagem. Para sistemas incapazes de rodar o SMTP, ou que não ficam ligados na rede o tempo todo, foram desenvolvidos ainda outros protocolos: POP (Post Office Protocol), atualmente na versão 3 [4] e IMAP [5] (Internet Message Access Protocol). Estes protocolos são encarregados de baixar as mensagens do servidor de email do destinatário para seu UA. A grande diferença entre o POP e o IMAP é a localização dos emails. No IMAP, o email fica no servidor, sendo apenas lido no UA. No POP, o email pode ser transferido para o cliente, o que torna a leitura a partir de múltiplos computadores menos confortável.

Devido ao sistema de transporte de correio eletrônico ter sido fixado em ASCII de 7 bits[1], as mensagens são enviadas como texto, mesmo se tiverem arquivos (sons, imagens, vídeos, planilhas, etc). Isso é possível graças ao MIME [6] - Multipurpose Internet Mail Extentions – que define codificações e o formato da mensagem de forma a ser possível converter arquivos binários em caracteres ASCII. Na codificação mais robusta, chamada de BASE64 [19], os caracteres que são usados para a codificação são A-Z, a-z, 0-9, + e -, que existem em todos alfabetos usados. Isto permite que, mesmo que o texto codificado seja "traduzido" para outra codificação (ASCII para EBCDIC, por exemplo), o conteúdo ainda possa ser recuperado.

No destinatário final o texto é reconvertido para o formato original.

Figura 1: Sistema de Transporte de Correio Eletrônico na Internet

III. PROBLEMAS COM O CORREIO ELETRÔNICO

Nesta seção apresentaremos alguns problemas associados ao correio eletrônico, e algumas soluções propostas para melhorar estas falhas, comparando com a modificação proposta no sistema de transporte.

A. SPAM

Pela grande facilidade de uso e relativa falta de segurança, o email tem sido usado por pessoas mal-intencionadas para difundir mensagens indesejadas – conhecidas como SPAM - e vírus e *worms*, que causam imenso prejuízo a pessoas e empresas. Além disso, o crescimento do volume de tráfego pode ocasionar grandes congestionamentos, com mensagens chegando atrasadas a seus destinatários, ou até mesmo serem perdidas. Mas, afinal, o que é SPAM?

O termo SPAM se refere ao envio, sem prévia solicitação e sem que os usuários desejem recebê-los, de emails comerciais, pirâmides e afins. Normalmente partem de poucos endereços (ou um só) e são copiados para milhares de destinatários.

Apesar de parecer óbvia, a distinção do que é ou não SPAM nem sempre é clara. Segundo a Mail Abuse Prevention System (MAPS) uma mensagem eletrônica é SPAM se [2]:

- A identidade e características pessoais do destinatário são irrelevantes pois o conteúdo da mensagem se aplica igualmente a várias pessoas.
- O destinatário não deu, deliberadamente, permissão explícita (que ainda pode ser revogada) para que fosse enviada.
- A transmissão da mensagem parece, para o destinatário, dar um benefício desproporcionalmente maior para o remetente.

[1] O SMTP já foi estendido para permitir o trânsito de informação com 8 bits [20]

Pode-se concluir pela definição acima, que o que é ou não SPAM, depende de uma interpretação pessoal de cada um (cada um deve decidir se o conteúdo da mensagem é ou não relevante e se a vantagem para o emissor é desproporcionalmente maior). Mas uma coisa é certa: o termo SPAM só faz sentido quando aplicado a mensagens enviadas para vários destinatários ("bulk mail"), pois, quando enviamos um email para alguém pela primeira vez, tecnicamente esta mensagem não foi explicitamente autorizada. O SPAM é um problema por várias razões:

- É difícil acabar com ele pois, muitas vezes, não se consegue localizar a origem. *Spammers* (pessoas ou empresas que enviam SPAM) utilizam emails fictícios ou criados especificamente para mandar uma carga de mensagens e depois desativam a conta. Como o email já está em curso, a desativação da conta não afeta a entrega do email.
- Consome recursos compartilhados. O tráfego de SPAM força os provedores de internet ("ISP") a ter de comprar mais largura de banda para poder suportar o aumento de volume de mensagens. Esse aumento de tráfego pode chegar a um ponto em que inviabilize a própria Internet.
- Penaliza duplamente o usuário. Como dito acima, o aumento do tráfego força os provedores a ter que comprar mais largura de banda. Esse custo adicional é repassado aos usuários – que já são vítimas do SPAM. Enquanto isso o custo para o *spammer* é muito baixo.
- Consome recursos privados. Se uma empresa tem 100 funcionários e estes passam 10 minutos por dia lendo e apagando SPAMs, no final de um mês o tempo total perdido pela empresa será de, aproximadamente, 360 horas de trabalho.
- Custo imensurável. Não se pode quantificar o custo da irritação e da frustração das pessoas ao abrirem seus emails e descobrirem várias mensagens de SPAM.

B. Vírus

Os vírus são programas que infectam a máquina de um usuário e que podem trazer todo tipo de problemas, desde simples inconvenientes (uma janela que fica aparecendo) até graves complicações, que necessitem que a máquina seja reformatada. A forma mais usual de combate aos vírus são os programas anti-vírus, que devem ser instalados e mantidos atualizados pelos próprios usuários.

C. Aumento do volume de tráfego na Internet

O aumento do volume de tráfego também é um grande problema (não só do email, mas da Internet em geral). Os nós intermediários (roteadores) nem sempre conseguem acompanhar o aumento na carga e acabam atrasando ou até mesmo descartando pacotes que não forem capazes de processar. Também o volume de mensagens nos servidores de email pode ser um problema.

D. Espaço de armazenamento das mensagens

Outro problema com o correio é a quantidade de dados a ser armazenada nos servidores. As caixas postais dos usuários se enchem rapidamente devido ao SPAM, e se a caixa tiver tamanho limitado (devido por exemplo a política de quotas do provedor), emails legítimos podem não ser recebidos porque a caixa de correio está cheia.

E. Soluções utilizadas atualmente

Atualmente o combate ao SPAM é feito, principalmente, de duas formas:

Filtros de SPAM: são programas que filtram todas as mensagens que são classificadas como SPAM. Esses filtros podem estar localizados na máquina do usuário ou em seu servidor de email. O problema com esse tipo de software é que muitas vezes emails legítimos acabam classificados como SPAM.

Alguns dos tipos de filtros são:

- Integrated, internet-based SPAM filters: são serviços oferecidos por terceiros e, geralmente, pagos. O banco de dados com as definições do que é SPAM é remoto e comum para todos os usuários. Isso gera um problema pois, o que é SPAM para uns pode não ser para outros. O número de falsos positivos tende a ser alto.
- Integrated, algorithmic SPAM filters: utilizam algorítimos para determinar o que é SPAM. Um dos mais usados é o algoritmo Bayesiano, que procura estatisticamente por palavras que aparecem em SPAMs e utiliza estas estatísticas para classificar os emails que são recebidos. Para se adaptar a isso os *spammers* têm trocado texto por imagens.
- Proxy SPAM filters: um proxy é um programa localizado entre a máquina do usuário e seu servidor de email. Ele pode estar localizado na própria máquina do usuário ou em seu servidor POP. Atualmente esses filtros são difíceis de se configurar em servidores IMAP, porque não há lugar bem definido para sua ação.
- Server-side SPAM filters: são filtros localizados no servidor de email. Dependendo da configuração podem apagar diretamente os emails classificados como SPAM ou apenas marcá-los como SPAM e deixar que o usuário lide com eles como preferir.

Algumas empresas (como a Microsoft, por exemplo), estão trabalhando em filtros inteligentes, que seriam capazes de aprender e acompanhar as mudanças de táticas dos *spammers*.

DNS Blacklist: monta uma lista de endereços de onde partem SPAMs e barra mensagens vindas desses endereços. Essas listas são produzidas e colocadas na Internet por organizações (DBSL.org [10], rfc-ignorant.org [11], entre outras) e utilizadas por vários ISP's. O grande problema com essas listas é que sites legítimos podem ser incluídos por engano – existem vírus como o MyDoom, por exemplo, que copiam os endereços IP das máquinas infectadas e enviam emails em nome delas - ou de forma mal intencionada, causando enormes prejuízos. A validade das listas está diretamente ligada à seriedade da organização, que deve atualizá-las com freqüência.

Algumas empresas estão desenvolvendo filtros adaptativos [7] e trabalhando em tecnologias que permitirão identificar o remetente dos emails, através da autenticação do remetente. Esta autenticação é feita da seguinte forma:[22]

1) Informação de autenticação para o servidor que envia email é colocada no DNS (ex: uma chave pública para assinatura)

2) o servidor que recebe email compara as referências do servidor que está enviando o email com a informação do servidor contida no DNS, e assim valida o servidor

Outros mecanismos para desencorajar o SPAM são a cobrança por email enviado e a criminalização do SPAM.

Outras pessoas (como Suzanne Sluizer – co-autora do MTP – Mail Transfer Protocol, o antecessor direto do SMTP) defendem que seja escrito um novo protocolo para eliminar as falhas de segurança do SMTP[8]. Um dos projetos é o IM2000[23]. Este caminho é similar a solução apresentada neste artigo. No entanto, como o SMTP é um padrão de enorme aceitação, acreditamos ser melhor mantê-lo o mais inalterado possível, e tentar corrigir os problemas advindos da sua falta de segurança através da limitação do que é transportado por ele.

Soluções para outros problemas do SMTP menos drásticas passam por estender suas funcionalidades. A RFC 3207, por exemplo, propõe uma extensão que permite que clientes e servidores SMTP usem TLS – Transport Layer Security para prover uma comunicação segura e autenticada na Internet[9].

As soluções descritas acima, apesar de válidas, atuam em pontos específicos, e, muitas vezes, dependem da colaboração de Governos (como criação de legislação pertinente), provedores (instalação de filtros) e até dos usuários (manter os seus antivirus atualizados). Nossa solução se difere delas por atacar os vários problemas de forma simples e integrada, e, uma vez implementada, funcionará por sí só, não requerendo manutenção periódica como filtros e bases de dados (e até legislação!). Além disso, apesar de não envolver nenhuma mudança no protocolo propriamente dito (SMTP), a solução proposta muda o modelo em que é utilizado, fazendo com que suas falhas de segurança sejam mais difíceis de serem exploradas.

IV. MODELO PROPOSTO

No modelo proposto (ver Figura 2) quando a mensagem chega ao servidor SMTP do remetente, é feita uma cópia do cabeçalho, adicionando-se mais algumas informações que permitirão que o destinatário veja a mensagem com segurança (ver subseção C), e somente esta cópia é enviada. A mensagem completa fica armazenada. O cabeçalho prossegue via SMTP, como no modelo atual, até chegar ao destinatário, para que este decida então, se deseja baixar a mensagem completa ou não. Caso afirmativo ele envia uma requisição HTTP ao servidor de armazenamento que encaminha a mensagem. Caso contrário ele pode apagar a mensagem diretamente no servidor de armazenamento ou deixá-la guardada para lê-la uma outra hora. Para viabilizar o modelo dois novos processos (*daemons)* foram desenvolvidos: um para dividir a mensagem (separar o cabeçalho do corpo) e enviar o cabeçalho para o destinatário e um outro para gerenciar a entrega do corpo. Este último transfere a mensagem quando esta é requisitada, faz o "coletor de lixo" para deletar mensagens expiradas e efetua a autenticação de segurança.

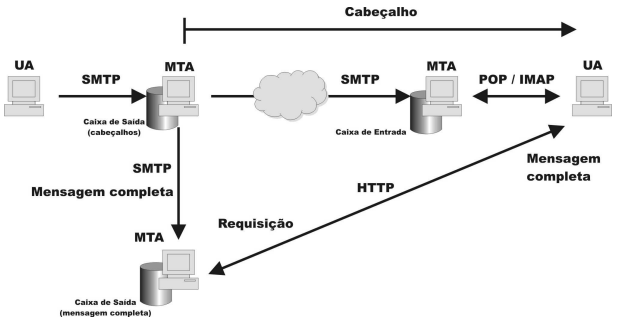

Figura 2: Sistema de Transporte proposto

A. Dois novos processos

Para fazer a divisão da mensagem, foi criado um novo processo que deve ser executado no MTA do remetente. Este processo faz uma análise da mensagem original, separando o cabeçalho do corpo. Ao cabeçalho ele adiciona o ID dessa mensagem, que é usado pelo destinatário para recuperar o corpo da mensagem. O mesmo ID é adicionado ao corpo. Este ID é composto de duas partes, uma que é o identificador de armazenamento, para possibilitar a recuperação da mensagem, e outro usado para criptografar o corpo da mensagem, conforme será explicado abaixo, para aumentar a privacidade dos emails armazenados. Este novo cabeçalho contém ainda, uma lista dos arquivos anexados (se houver) e um link HTTP para o corpo da mensagem. O novo cabeçalho é enviado pelo MTA, enquanto o corpo é armazenado.

Outro servidor é responsável por entregar o corpo da mensagem quando este é solicitado. O processo que implementa o servidor é executado no repositório distribuído, que pode ser uma máquina distinta do MTA do remetente ou não. Ele faz a autenticação do destinatário através do ID da mensagem, de modo que este seja o único a ter permissão para ver o conteúdo da mensagem. Para evitar o acúmulo de mensagens, muitas vezes esquecidas pelos destinatários, este processo também executa um "coletor de lixo", que apaga mensagens após um certo tempo. Ao serem enviadas, as mensagems ganham um "prazo de validade". Após este prazo elas são apagadas. Somente as mensagens não lidas são afetadas por este coletor, uma vez que as mensagens lidas são apagadas assim que requisitadas.

B. Arquivos Anexados

Os arquivos anexados são listados no cabeçalho que vai para o destinatário. O corpo da mensagem, contém um link para o(s) arquivo(s), para que estes possam ser vistos.

Os arquivos são reconvertidos ao formato original no servidor de armazenamento. Quando é feita a requisição HTTP eles são enviados e exibidos no browser ou no programa apropriado (ou então o usuário pode optar por fazer o download do arquivo), conforme já acontece hoje quando é feita a solicitação de um arquivo a um servidor HTTP.

C. *Segurança*

Para garantir que somente o usuário a quem a mensagem se destina possa lê-la, um esquema de autenticação foi desenvolvido. Ao ser feita a divisão da mensagem um identificador é criado e gravado tanto na mensagem quanto no cabeçalho, utilizando um esquema de criptografia. Ao clicar no link para ver a mensagem, este identificador é retornado e somente com ele a mensagem pode ser lida. Para aumentar a segurança do email, é possível usar um esquema de chave privada/chave pública. Se o remetente conhecer a chave pública do destinatário, a mensagem é primeiro criptografada usando a chave pública do destinatário, e depois a parte do identificador da mensagem é usada para a criptografia. O identificador é um número de 512 bits escolhido aleatóriamente, de forma a dificultar ataques a base de dados (onde um atacante tentaria ler mensagens pedindo mensagens em sequência).

D. *Envio do cabeçalho*

Após ser feita a divisão da mensagem, o cabeçalho (com os campos adicionais) segue o caminho convencional de um email hoje, isto é, segue via SMTP pela rede até o MTA (servidor SMTP) do destinatário e deste para o servidor POP ou IMAP, de onde será transferido pelo usuário para sua máquina. Como o cabeçalho normalmente será bem menor que a mensagem completa, isto permite o envio do mesmo a terminais com memória limitada, como telefones celulares e PDAs.

E. *Recuperação do corpo da mensagem*

Para ver o corpo da mensagem o usuário deve clicar no link que acompanha o cabeçalho, enviando uma requisição HTTP para a máquina onde está armazenada a mensagem. O identificador é usado para localizar a mensagem e para descriptografá-la. Assim, não é necessária nenhuma alteração nos clientes sendo usados atualmente, já que estes, em grande parte, já possuem suporte a *links* http.

F. *Listas*

Muitos emails são endereçados a mais de uma pessoa, na forma de listas de discussão. A maior parte dos sistemas de correio permitem a criação de listas, facilitando o envio de email para grupos de usuários. No modelo proposto usamos a mesma solução do SMTP, que é criar uma cópia da mensagem para cada nome da lista. Uma solução alternativa seria manter uma única cópia da mensagem e usar um banco de dados para gerenciar quem já leu a mensagem, e quando ela poderia ser apagada. Apesar disto economizar espaço de armazenamento no servidor, é uma solução mais complexa, que pode ser implementada sem alterar o resto do modelo.

G. *Vantagens do novo modelo*

O modelo proposto apresenta as seguintes vantagens:

- **Dificulta o SPAM**: o novo modelo dificulta a prática de SPAM, pois somente o cabeçalho das mensagens é automaticamente enviado ao destinatário. As mensagens completas somente serão vistas se o usuário quiser. Além disso o novo modelo transfere o ônus da armazenagem das mensagens para o servidor do remetente - fica mais difícil e mais caro armazenar milhares de mensagens em seu servidor de saída.
- **Diminui o tráfego na rede**: como somente um pequeno cabeçalho é enviado, o volume de tráfego na Internet será reduzido.
- **Caixa de entrada distribuída:** a caixa de entrada do usuário fica distribuída entre vários servidores distintos, ao contrário do IMAP, em que todas as mensagens ficam em um mesmo servidor. Dessa forma as mensagens podem ser lidas de qualquer máquina e, caso aconteça algo com um dos servidores, as mensagens nos outros servidores ainda estarão disponíveis.
- **Diminui o risco de infecção por vírus**: ao ler o cabeçalho o usuário tem uma chance de identificar ou não o remetente, enquanto o arquivo (possivelmente) infectado ainda está no servidor. Com isso ele pode decidir não buscar a mensagem ou então ler a mensagem, mas não abrir o anexo. Se o usuário receber um email com virus, o servidor que originou este email é facilmente identificado, e pode ser contactado para impedir que outras pessoas recebam o virus. Apesar disto ainda permitir que servidores fantasmas existam, a necessidade de criar uma presença mais permanente do que simplesmente para o envio de email aumenta o custo do SPAM para o remetente.
- **Compatibilidade**: como o modelo proposto utiliza os protocolos SMTP e HTTP, ele é totalmente compatível com o modelo atual e, portanto, pode ser implementado em etapas, sem que os serviços de email em funcionamento hoje tenham que ser interrompidos.

H. *Desvantagens do modelo apresentado*

O modelo não termina completamente com o *SPAM*. Servidores podem ser criados para a distribuição de email, da mesma forma que pornografia é colocada em servidores de acesso livre da Internet. Como o modelo pode conviver com o antigo, também *spammers* podem usar o SMTP até que seja dado aos usuários ferramentas para só aceitar email se este for do tipo "somente cabeçalho". O uso de HTTP permite que virus e *worms* que explorem as falhas de segurança dos leitores de HTML continuem a ser usados.

O modelo também gera uma carga maior nos servidores de envio de email, que agora também tem a função de armazenar as cartas até que sejam lidas. O "prazo de validade", apesar de reduzir o problema anterior, também pode ser outro problema, porque usuários podem perder emails se sairem de férias por períodos longos, por exemplo. Tem de ser criado um mecanismo similar ao auto-respondedor *"vacation"*, que permita que o usuário mantenha suas mensagens ativas até ele retornar (ou então receba todos os emails ou os emails que estão para expirar enquanto o usuário está de férias).

Finalmente, nem todos os usuários podem gostar de ter mais um passo entre receber a notificação de email e de receber o corpo do email. Se o usuário está num sistema com banda limitada (ex: modem discado), o ganho de não ter que

transferir todos os emails pode ser menor do que o tempo requerido para transferir cada email desejado. Além disto, o mecanismo de filtragem de SPAM pode requerer acesso ao corpo do email. Se buscar o corpo for uma tarefa manual, isto pode tomar mais tempo que receber todos os emails, com SPAM ou não. Assim, é necessário criar métodos automatizados que se adequem as necessidades dos usuários. Acreditamos que isto pode ser feito numa segunda etapa através da modificação dos programas de leitura de email.

O objetivo original deste modelo era racionalizar o transporte de email, evitando o envio de emails com conteúdo pesado para a caixa de correio do usuário, que é desvantajoso principalmente quando o usuário está conectado por um enlace de baixa capacidade, como uma linha discada. Isto se torna cada vez mais importante pois o email está tomando o lugar do FTP, e cartas com conteúdo multimídia (fotos, arquivos executáveis) se tornam cada vez mais comuns. Assim, ele é ortogonal as outras iniciativas de término de SPAM, e pode ser adotado juntamente com elas.

I. Estágio atual

O protótipo de testes encontra-se em estado avançado. Ele foi escrito em JAVA para ser multi-plataforma. Atualmente, o protótipo funciona da seguinte forma: para cada mensagem que chega ao servidor uma nova thread é iniciada, que executa a operação de criação da nova mensagem com o cabeçalho e os dados adicionais, e a encaminha ao servidor SMTP para que esse a envie através da Internet. O corpo é gravado num diretório em uma árvore de diretórios indexada por data para facilitar a remoção de mensagens, que agora tem tempo de validade fixo de um mês. Para o serviço de entrega do conteúdo (corpo) das mensagens foi escrito um pequeno servidor HTTP, que também utiliza threads, para poder atender a várias requisições simultâneas.

Ainda não está sendo feita a criptografia das mensagens, e está sendo estudado se a melhor forma de implementar o servidor de conteúdo das mensagens é através de scripts CGI para servidores de HTTP de uso geral, como o Apache [21]. A vantagem desta solução é que estes servidores são robustos e testados. No entanto, se faz necessário escrever scripts para cada família de servidores.

J. Próximos passos

Assim que for adicionada a criptografia, começaremos os testes colocando o servidor para funcionar espelhando o email do laboratório. Isto permitirá os testes de validação. Após estes testes, serão feitos testes de carga, para analizar o desempenho do sistema quando estressado. Se for usada a solução de scripts CGI para a entrega do conteúdo das mensagens, a maior sobrecarga será no envio de mensagens para listas grandes, o que não deve oferecer problemas já que não tem requerimentos de velocidade.

V. CONCLUSÃO

O correio eletrônico foi a primeira ferramenta da Internet que alcançou penetração universal, e é, ao lado da Web, a aplicação mais usada atualmente. Infelizmente, a facilidade de uso, e a própria natureza da aplicação a torna vulnerável a abusos como o SPAM ou à disseminação de programas maliciosos como vírus e *worms*. No entanto, acreditamos que o modelo de transporte pode ficar mais seguro se impossibilitarmos que um servidor pirata injete milhares de emails e saia da Internet sem deixar vestígios. Requerendo uma presença constante, servidores que estão sendo usados de forma maliciosa podem ser identificados, e gestões feitas contra seu mal uso.

O modelo de transporte de email proposto neste artigo segue os padrões vigentes atualmente, e é perfeitamente compatível com os programas clientes usados hoje. As únicas modificações necessárias são nos servidores. Mesmo estes usam protocolos já definidos, SMTP e HTTP. Servidores novos podem co-existir com servidores que seguem o modelo antigo, o que permite uma migração gradual, que é a única possível devido a natureza distribuída da administração da Internet.

Apesar deste modelo não terminar automaticamente com o SPAM, acreditamos ser uma ferramenta poderosa para minimizá-lo, e com um custo menor em termos de processamento requerido no servidor e maior facilidade de implementação que os modelos propostos atualmente.

REFERÊNCIAS:

[1] Darwin Magazine - http://www.darwinmag.com/read/010102/buzz_mover.html.

[2] maps - http://www.mail-abuse.com/spam_def.html

[3] RFC 821 - Simple Mail Transfer Protocol, J. B. Postel, Agosto 1982

[4] RFC 1939 - Post Office Protocol - Version 3, J. Myers, C. Mellon, M. Rose, Maio 1996

[5] RFC 3501 - Internet Message Access Protocol - Version 4rev1, M. Crispin, Março 2003

[6] RFC 1521 - MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies, N. Borenstein, N. Freed, Setembro 1993

[7] Microsoft Executive E-Mail: Toward a Spam-Free future - http://www.microsoft.com/mscorp/execmail/2003/06-24antispam.asp

[8] News.com - http://news.com.com/2100-1038_3-5058610.html

[9] RFC 3207 - SMTP Service Extension for Secure SMTP over Transport Layer Security, P. Hoffman, Fevereiro 2002

[10] http://dsbl.org/main/

[11] http://www.rfc-ignorant.org/

[12] http://en.wikipedia.org/wiki/Fidonet

[13] http://www.fidonet.org/

[14] http://www.fidonet.org/inet92_Randy_Bush.txt

[15] http://en.wikipedia.org/wiki/Usenet

[16] Grier, David Alan; A Social History of Bitnet and Listserv, 1985–1991; 2002 http://www.computer.org/annals/articles/bitnet.htm

[17] http://en.wikipedia.org/wiki/Webcasting

[18]Luiz Magalhães, "Correio Eletrônico em Português", Tese de Mestrado, PUC-Rio 1994

[19] RFC 3548 - The Base16, Base32, and Base64 Data Encodings, S. Josefsson, Julho 2003

[20] RFC 2821 Simple Mail Transfer Protocol, J. Klensin, Abril 2001

[21] http://www.apache.org/

[22]http://www.ip97.com/iij_introduces_sender_authentication_fdc.aspx

[23] http://www.magma.com.ni/~jorge/im2000/im2000.html