Documentos

Março, 2005 50

ISSN 1677-9274

Mecanismos contra Spams para Servidor de Correio Eletrônico

Embrapa

ISSN 1677-9274
Março, 2005

Empresa Brasileira de Pesquisa Agropecuária
Embrapa Informática Agropecuária
Ministério da Agricultura, Pecuária e Abastecimento

Documentos 50

Mecanismos contra Spams para Servidor de Correio Eletrônico

Marcelo Gonçalves Narciso

Campinas, SP
2005

Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)
Av. André Tosello, 209
Cidade Universitária "Zeferino Vaz" – Barão Geraldo
Caixa Postal 6041
13083-970 – Campinas, SP
Telefone (19) 3789-5743 – Fax (19) 3289-9594
URL: http://www.cnptia.embrapa.br
e-mail: sac@cnptia.embrapa.br

Supervisor editorial: *Ivanilde Dispato*
Normalização bibliográfica: *Maria Goretti Gurgel Praxedis*
Editoração eletrônica: *Área de Comunicação e Negócios (ACN)*

1ª. edição on-line - 2005

Narciso, Marcelo Gonçalves
Mecanismos contra spams para servidor de correio eletrônico / Marcelo Gonçalves Narciso. – Campinas : Embrapa Informática Agropecuária, 2005.
28 p. : il. – (Documentos / Embrapa Informática Agropecuária ; 50).

ISSN 1677-9274

1. Serviço de correio eletrônico. 2. Spam. 3. Combate a spam. I. Título. II. Série.

CDD – 004.692 21st. Ed.

Autor

Marcelo Gonçalves Narciso
Doutor em Computação Aplicada, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP
e-mail: narciso@cnptia.embrapa.br

Apresentação

O serviço de correio eletrônico tem sido uma forma de se propagar mensagens comerciais a custo zero. Mensagens não solicitadas, relativas a propagandas sobre remédios, softwares, etc. têm se propagado em larga escala pela internet e têm trazido sérios transtornos a usuários por todo o mundo. Estas mensagens são conhecidas como spams, isto é, envio de mensagens não solicitadas, em grande número, a destinatários desconhecidos.

Além de ser desagradável para o usuário ter sua caixa postal cheia de spams, esses sobrecarregam o servidor de correio eletrônico, podendo causar lentidão no serviço de entrega de mensagens. O tráfego de dados pela rede também é prejudicado, pois as mensagens são enviadas para um grande número de pessoas e isto pode congestionar a própria internet. Desta forma, é importante que o servidor de correio eletrônico tenha um mecanismo de rejeição de spams .

Assim, este trabalho descreve um conjunto de opções para se reter spams em um servidor. A vantagem é que, antes que a mensagem chegue ao usuário, o servidor analisa a mensagem e, caso seja constatado que a mensagem seja um spam, descarta a mensagem. Desta forma, a qualidade do serviço de mensagens oferecido ao usuário aumenta, bem como diminui o tráfego de mensagens na rede.

Espera-se que este trabalho seja útil para quando da configuração de servidor de correio eletrônico de tal forma que impeça a disseminação de spams para os usuários.

José Gilberto Jardine
Chefe-Geral

Sumário

Mecanismos contra Spams para Servidor de Correio Eletrônico

Marcelo Gonçalves Narciso

Introdução

O serviço de mensagens é um dos mais importantes em uma rede. Desta forma, medidas de segurança relativas a este serviço são de suma importância. Mensagens indesejadas, tais como propaganda, em grande quantidade por dia, têm sido a preocupação de muitos administradores de redes e de sistemas no sentido de combatê-las. Estas mensagens são conhecidas por spam.

Spam é o termo pelo qual é comumente conhecido o envio de mensagens eletrônicas não solicitadas a uma grande quantidade de pessoas de uma vez, geralmente com cunho publicitário, mas não exclusivamente.

O repúdio ao spam na rede não surge gratuitamente, mas sim graças a fatores como: a perturbação, irritação e mau humor das vítimas; prejuízo causado com o desperdício de recursos que vão, desde o tempo gasto pelos milhões de usuários em limpar suas caixas postais todos os dias, até o tempo gasto pelos administradores, grupos de combate a spams e grupos de segurança em tentar de alguma maneira coibir tal ato, culminando no desperdício e, em muitos casos, degradação de desempenho de servidores e da rede. Atualmente, spams têm trazido links que, na verdade, são downloads de vírus, o que é extremamente perigoso, principalmente quando os usuários, por curiosidade, acessam o link.

Para combater spams, faz-se necessário um conjunto de mecanismos, tanto na máquina servidora de correio eletrônico quanto nas máquinas clientes.

Este trabalho visa a descrever mecanismos de defesa (rejeição de spams) no computador responsável pelo serviço de mensagens (servidor de correio eletrônico) quando este recebe spam.

Será considerado neste artigo o MTA (Mail Transfer Agent) sendmail (Sendmail, 2004), visto que é um dos mais usados na internet. Porém, as sugestões a serem apresentadas neste trabalho servirão, com as devidas adaptações, para outros softwares MTA, tais como postfix (Postfix, 2004) ou qmail (Qmail, 2004), por exemplo.

Sendmail

Para um melhor entendimento do processo de instalação e configuração de *softwares* que combatem spams, faz-se necessário discorrer um pouco sobre o sendmail (Sendmail, 2004), *software* largamente empregado em sites para a entrega e recebimento de mensagens pela internet e rede local.

Quando um usuário qualquer (beltrano@site.com.br, por exemplo), a partir de sua máquina da rede local, envia mensagem para outro usuário (fulano@site1.gov.br, por exemplo), o processo do sendmail que fica na servidora da rede, cujo domínio é site.com.br, deverá ler o endereço, determinar se é local ou não, e então enviar a mensagem para o endereço correto (beltrano@site1.gov.br). Para isto, o sendmail tem uma série de regras, as quais estão cadastradas em seu arquivo de configuração, conhecido por sendmail.cf. Neste arquivo podem ser configurados uma série de parâmetros, tais como: diretórios onde ficam os arquivos de aliases da rede, time out de conexão, tempo limite para entregar uma mensagem, domínio da rede local, mecanismos para combater spam, etc. As regras que ficam no sendmail.cf têm o objetivo de fazer uma verificação no endereço eletrônico para onde vai a mensagem e escolher a forma como a mensagem vai ser enviada (se localmente ou para um *site* da internet). Além disso, o sendmail.cf também tem regras para verificação do endereço de quem enviou a mensagem. Isto é útil para se combater spam, pois conforme o endereço, a mensagem poderá ser rejeitada ou não. A Fig. 1 ilustra o funcionamento do sendmail.

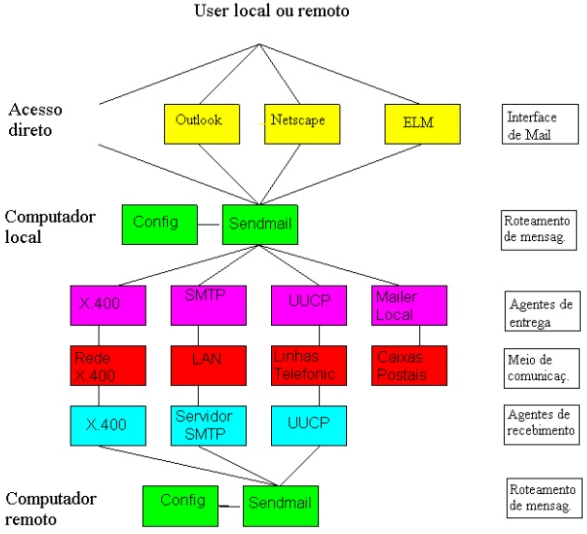

FIG. 1. Esquema do funcionamento do sendmail.

A respeito da forma de entrega de mensagens, se esta tiver destinatário com endereço da rede local, o sendmail escolhe o entregador de mensagens locais (usualmente são programas executáveis ou *scripts*, e os mais comuns são mail, mail.local, procmail, ou scanmails) e envia a mensagem para o destinatário da rede local. Se a mensagem estiver endereçada para um usuário fora da rede local, o sendmail, através de uma série de rotinas (funções) que vem em seu pacote, envia a mensagem conforme os parâmetros estabelecidos no seu arquivo de configuração, o sendmail.cf (Costales & Allman, 1997).

Uma vez colocado um pequeno resumo sobre o sendmail, pode-se então discorrer sobre a instalação de mecanismos de se evitar spam (ou mensagens indesejáveis) a partir da estação servidora de mensagens da rede local. Os mecanismos são referenciados a partir do sendmail.cf. Quando uma mensagem chega a algum destinatário da rede local, o sendmail ativa o entregador de mensagens locais, o qual pode ser o procmail ou scanmails, conforme o que estiver configurado em sendmail.cf. Caso o entregador de mensagens locais seja o scanmails (Scanmails, 2004), este deverá chamar o executável procmail (Procmail, 2004). O scanmail serve para verificar presença de vírus anexados a mensagens. Maiores detalhes sobre scanmails podem ser vistos em Narciso (2001).

Nos tópicos a seguir serão descritas formas de se barrar spams a partir de listas negras feitas localmente ou a partir de sites de listas negras. Além disso, formas de se criar listas negras ou brancas a partir do arquivo procmailrc e como chamar *softwares* antispams também serão enfocadas.

Bloqueio de Spams através de Acesso a Sites de Lista Negra

Uma lista negra é um banco de dados que contém *sites*, endereços eletrônicos e endereços IPs considerados fontes de spams. O servidor de correio eletrônico consulta o *site* que oferece o serviço de lista negra e verifica se o remetente da mensagem está cadastrado na lista negra. Se não estiver, o servidor continua o envio da mensagem. Se estiver, a mensagem é descartada e o remetente é notificado.

Outros IPs que podem estar na lista negra são relativos a relay. Um servidor de correio eletrônico é um relay quando ele recebe uma mensagem de um remetente fora de seu domínio e envia a mensagem para um outro domínio (diferente do domínio do relay). Quando o servidor está mal configurado, pode ser usado para propagar spams. Os servidores com "relay aberto" permitem a autenticação de qualquer endereço eletrônico que passe por seu sistema e não apenas daqueles que fazem parte dos domínios gerenciados pelo servidor. Desta forma, a origem real da mensagem pode ser mascarada.

Nos primeiros anos de operação da internet, permitir relay de terceiros era necessário e aceito como maneira de rotear mensagens, mas tal necessidade foi suplantada pelos avanços tecnológicos.

Existem *sites* de listas negras conhecidas mundialmente tais como www.mail-abuse.org, spam.abuse.net, www.abuse.net, www.cauce.org, etc.

Existe também uma lista negra brasileira de relays, a qual está no *site* http://www.globalmedia.com.br/orbl/. Um dos principais objetivos é oferecer um suporte mais adequado do que o prestado por listas negras estrangeiras aos adminstradores de redes brasileiros, evitando situações comuns em que usuários são indevidamente bloqueados.

Para que o sendmail possa consultar a lista negra para verificar se o remetente está ou não na lista negra, é necessário que o arquivo de configuração sendmail.cf tenha sido gerado com FEATURE(`dnsbl'). Detalhes sobre este comando e outros a serem usados para gerar o arquivo sendmail.cf estão em Sendmail (2004).

O arquivo sendmail.cf é gerado a partir de um outro arquivo, de extensão .mc. Suponha que o arquivo de extensão .mc seja sendmail.mc. Para gerar o arquivo sendmail.cf, basta executar o comando

m4 sendmail.mc > sendmail.cf

m4 é um arquivo executável que vem no pacote de instalação do sendmail. Um exemplo de arquivo .mc (no exemplo, sendmail.mc) que vai originar o arquivo sendmail.cf é o que está a seguir:

```
Divert(-1)
divert(0)dnl
VERSIONID(`$Id: cs-solaris2.mc,v 8.12 1999/02/07 07:26:00 gshapiro Exp $')
OSTYPE(linux)dnl
DOMAIN(generic)dnl
MASQUERADE_AS(`cnptia.embrapa.br')dnl
MASQUERADE_DOMAIN(`cnptia.embrapa.br')dnl
FEATURE(`relay_based_on_MX')dnl
FEATURE(`masquerade_entire_domain')dnl
FEATURE(`masquerade_envelope')dnl
FEATURE(use_cw_file)dnl
FEATURE(access_db, hash  -o /etc/mail/access)
FEATURE(mailertable, hash  -o /etc/mail/mailertable)
FEATURE(virtusertable, hash  -o /etc/mail/virtusertable)
FEATURE(dnsbl)
FEATURE(dnsbl,`blackholes.mail-abuse.org',` Mail from $&{client_addr}rejected,
see http://mail-abuse.org/cgi-bin/lookup?$&{client_addr}')dnl
FEATURE(dnsbl,`relays.mail-abuse.org',` Mail from $&{client_addr}rejected; see
http://mail-abuse.org/cgi-bin/nph-rss?$&{client_addr}')dnl
FEATURE(dnsbl,`dialups.mail-abuse.org',` Mail from dial-up rejected; see http://
mail-abuse.org/dul/enduser.htm')
MAILER(local)dnl
MAILER(smtp)dnl
define(`confCW_FILE',`-o /etc/mail/sendmail.cw')dnl
define(`confPRIVACY_FLAGS', ``authwarnings noexpn novrfy needmailhelo'')
```

As opções que irão ser usadas no acesso a lista negra estão em negrito no arquivo sendmail.mc. Estas opções irão direcionar o sendmail a acessar as listas para consulta.

Exemplos de arquivos .mc vêm no pacote de instalação do sendmail. Mais informações sobre como gerar arquivos sendmail.cf a partir de algum arquivo .mc podem ser vistas em Sendmail (2004).

Bloqueio de Spams através de uma Base de Dados Local (Access)

Existem casos em que mensagens, aparentemente legítimas mas na verdade é um spam, não são barradas pelos filtros (spamassassin, procmail, etc). Neste caso, conhecendo-se o domínio de onde a mensagem vem, ou o endereço eletrônico do remetente, pode-se rejeitar a mensagem, caso mensagens indesejadas sejam continuamente enviadas por este domínio. Esta é uma das características que o sendmail (ou postfix) suporta. Para implementar esta característica, basta configurá-la no arquivo sendmail.cf e executar alguns procedimentos. Os passos seriam os seguintes:

1. Contruir o sendmail.cf usando a feature acess_db (FEATURE(access_db)). Mais detalhes podem ser vistos em (Sendmail, 2004). No arquivo sendmail.cf irá aparecer a seguinte linha

Kaccess hash -T<TMPF> -o /etc/mail/access

2. Inserir no arquivo /etc/mail/access os domínios ou endereços eletrônicos. Segue um exemplo de conteúdo deste arquivo:

```
200.150.59 RELAY

meuspam.com.br      550      Nao permitimos Spammers

spam.com.br REJECT
fulano.spam.com.br OK
```

Nessa configuração foi permitido o RELAY da rede para 200.150.59. Mensagens provenientes de spam.com.br serão rejeitadas, menos aquelas provenientes da estação fulano.spam.com.br. Mensagens provenientes de meuspam.com.br (550 = REJECT) serão rejeitadas e o remetente receberá a mensagem de erro "Nao permitimos Spammers". As opções que podem ser usadas no arquivo *access* são:

DISCARD - descarta a mensagem sem enviar um mensagem de erro a quem a enviou.
OK - permite que mensagem vinda de uma determinda origem seja aceita.
REJECT - rejeita a mensagem e retorna uma mensagem de erro para quem a enviou.
RELAY - permite determinado usuário utilizar o servidor para enviar mensagens.

3. Construir a base de dados access.db(/etc/mail/access.db)

```
# makemap hash access < access
```

Não haverá destruição do access original. O trecho do comando access < access significa que será gerado um arquivo access.db a partir de access. O executável makemap vem no pacote do sendmail e deverá estar sempre atualizado.Após esta

Opção estar configurada, o sendmail verifica se o remetente da mensagem (ou o *site* de onde ela provém) está no arquivo /etc/mail/access.db. Cada vez que o arquivo /etc/mail/access for atualizado, o comando makemap, tal como mostrado no passo 3, deverá ser executado.

A opção do arquivo access é eficiente quanto o spamassassin ou bogofilter ou procmail não conseguir barrar spams de um certo endereço ou *site*. Além disso, pode ser usado para barrar *sites* ou endereços indesejáveis temporariamente ou permanentemente. Na verdade, certos *sites* enviam muitos spams, e quando barrados usando-se o arquivo access retém mais spams do que o spamassassin ou filtros do procmail. Por exemplo, foi constatado que sites do tipo dsl.YYY.net.br, dsl.ZZZ.net.br e outros similares, quando colocados em uma lista negra, trouxe uma sensível diminuição da quantidade de spams. ZZZ e YYY são empresas conhecidas. *Sites* do domínio .biz também, quando barrados, retém grande quantidade de spams.

Para se ter uma noção da importância deste mecanismo de lista branca interna, observe a Tabela 1.

Tabela 1. Resultados de mecanismos de retenção de spams em abril de 2004.

Dia/abril	*Lista Negra*	*Endereços falsos*	*Spamassassin*
20	2.003	6.813	955
21	2.522	4.950	692
22	2.797	5.590	869
23	2.838	5.712	1.004
24	2.800	5.303	644
25	2.442	4.773	627
26	2.365	5.304	831
27	2.414	5.323	878
28	2.838	5.712	1.004

Na Tabela 1 pode-se constatar que, das mensagens rejeitadas, a lista negra (access) tem uma razoável parcela de contribuição, daí sua importância no combate aos spams.

Procmail

Conforme visto no item anterior, o Procmail é um entregador de mensagens locais. Para instalar o Procmail na máquina servidora de correio eletrônico, é necessário que nesta esteja instalado um compilador C (gcc, por exemplo). O servidor de correio eletrônico deverá então usar o procmail para entrega local das mensagens. Em plataformas Linux, Solaris, AIX, HP e outros sistemas Unix, outros entregadores de mensagens locais são definidos inicialmente. Porém, basta

instalar o Procmail e defini-lo como entregador de mensagens locais no sendmail.cf, para o caso do sistema estar usando o sendmail (ver www.sendmail.org). No arquivo sendmail.cf, o trecho do programa onde ficará o procmail é o seguinte:

```
Mlocal,   P=/usr/bin/procmail,          F=SAw5:/|@glDFMPhsfn,
S=EnvFromL/HdrFromL, R=EnvToL/HdrToL,
          T=DNS/RFC822/X-Unix,
          A=procmail -Y -a $h -d $u
```

A instalação do procmail (versao 1.1, usada neste exemplo, ou superior) pode ser obtida através do site (Procmail, 2004).

Conforme mencionado anteriormente, o arquivo procmailrc pode conter regras para rejeitar spams ou ainda chamar softwares para reconhecer e rejeitar spams. Segue um exemplo de regras para rejeitar spams, dado que são conhecidas palavras que podem aparecer no campo Subject ou corpo da mensagem.

```
#A regra abaixo rejeita toda mensagem que tiver Subject new photos from my party

:0

* ^Subject:.*new photos from my party
| /etc/procmail.d/recusaPalavra.pl
```

No exemplo citado, se a frase "new photos from my party" estiver no campo Subject da mensagem, é executado um script que envia uma mensagem ao remetente dizendo que a mensagem foi recusada. O *script* recusaPalavra.pl, foi feito em perl (PERL, 2004), mas o leitor poderá fazer seu próprio *script* para isto. A notificação de recusa da mensagem não é entregue ao destinatário, pois não é necessário e faria o mesmo papel de um spam.

```
# A regra abaixo rejeita toda mensagem que tiver Subject  Re: Thank you!
:0
* ^Subject:.*Re: Thank you!
| /etc/procmail.d/ recusaPalavra.pl

#A regra abaixo rejeita todo arquivo que tiver a frase I just wanted to share with you
#no corpo do texto  e envia a mensagem para /dev/null, eliminando a mesma.
:0 HB
* B ?? I just wanted to share with you
/Dev/null

#A regra abaixo rejeita todo arquivo que tiver a palavra viagra no corpo do texto
#
* B ?? Viagra
/dev/null
```

Este procedimento, entretanto, é válido quando se conhecem algumas palavras que aparecem em Subject ou no corpo da mensagem. É útil quando o software antispam não consegue barrar a mensagem.

Por outro lado, é possível fazer listas brancas para determinados *sites* ou usuários, por exemplo. Assim, se a mensagem vier do *site* local (cnptia.embrapa.br) por exemplo, cujo ip de qualquer máquina começa por 200.0.70, ou 192.207.194 ou 127.0.0.1, tem-se a seguinte regra para enviar a mensagem rapidamente, sem passar por outras regras ou filtros:

```
:0
* ^From:.*cnptia\.embrapa\.br
 {
    :0
    * ^Received: from[ ].*\[200\.0\.70\.
    $DEFAULT

    * ^Received: from[ ].*\[192\.207\.194\.
    $DEFAULT

    * ^Received: from[ ].*\[127\.0\.0\.1
    $DEFAULT

 }
```

Se um usuário ciclano@rrr.yyy.uu recebe mensagem de fulano@xxx.yyy.cd, pode ser feito um filtro da seguinte maneira:

```
:0
* ^From:.*fulano\@xxx\.yyy\.cd
* ^To:.*ciclano\@rrr\.yyy\.uu
$DEFAULT
```

Se quiser descartar mensagens que vêm de *sites* supostamente local (cnptia.embrapa.br, por exemplo) mas não é, pode ser feito o seguinte:

```
:0
* ^From:.*cnptia\.embrapa\.br
* !^Received: from[ ].*\[200\.0\.70\.
* !^Received: from[ ].*\[192\.207\.194\.
* !^Received: from[ ].*\[127\.0\.0\.1
* !Message-ID:.*cnptia.embrapa.br
 /Dev/null
```

Observe que se a mensagem diz ser proveniente de usuário do cnptia, mas o ip da estação de onde a mensagem foi enviada não corresponde aos possíveis ips do domínio cnptia.embrapa.br e o parâmetro Message-ID não provém do domínio cnptia.embrapa.br, então a mensagem deverá ser descartada.

Existem várias maneiras de se fazer listas negras ou brancas em procmailrc. Embora sejam de pouca divulgação os exemplos de como fazer estas listas, daí os exemplos anteriores, é possível aprender sobre regras em (Procmail, 2004)

O procmail tem outras aplicações interessantes, tais como barrar mensagens anexadas a mensagens, que podem ter vírus (Narciso, 2001). Além disso, pode ser configurado para ativar softwares antispam, tais como o spamassassin (Spamassassin, 2004) e o TMDA (TMDA, 2004).

Spamassassin

Um dos *softwares freeware* mais usados na internet para combater spams é o spamassassin. Este *software* analisa o corpo da mensagem (faz análise do texto) e pode verificar se o remetente da mensagem ou *site* de onde a mensagem provém está em uma lista negra (consulta *sites* tais como mail-abuse.org, ordb.org). Para isto, existe um conjunto de regras, que vêm com o *software* ou estão em outros *sites* de colaboradores (por exemplo, o arquivo br_rules.cf, a ser considerado mais a frente). Este antispam é conhecido por "baseado em regras". Porém, recentes versões têm incorporado também as características de filtros bayesianos, que se adaptam as mudanças que os spams sofrem com o tempo.

O pacote do spamassassim, bem como sua instalação, pode ser obtido através do *site* (Spamassassin, 2004)

Este *software* é chamado através do procmail desde que esteja configurado no arquivo procmailrc. Um exemplo de como configurar o procmailrc para invocar o spamassassin está descrito a seguir.

```
# SpamAssassin sample procmailrc
#
# The condition line ensures that only messages smaller than 250 kB  (250 * 1024
= 256000
# bytes) are processed by SpamAssassin. Most spam isn't bigger than a few k
# The lock file ensures that only 1 spamassassin invocation happens at 1 time, to
keep the
# load down.
#
:0fw: spamassassin.lock
#* < 256000
 /export/home/perl/bin/spamassassin -l /export/home/perl/logSpam

# Mensagens with a score of 15 or higher are almost certainly spam (with 0.05% false
positives #according to rules/STATISTICS.txt). Let's put them in a # different mbox.

:0 c:
* ^X-Spam-Flag: YES
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
/etc/procmail.d/recusaSpam.pl

:0:
* ^X-Spam-Flag: YES
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
/Export/home/perl/almost-certainly-spam

# All mail tagged as spam (eg. with a score higher than the set threshold)
# is moved to "probably-spam".
:0 c:
* ^X-Spam-Status: Yes
/export/home/perl/probably-spam

:0:
* ^X-Spam-Status: Yes
/etc/procmail.d/recusaSpam.pl
```

Observe que o arquivo escrito em perl /export/home/perl/bin/spamassassin é invocado toda vez que o procmail acessar uma mensagem para entregar. Este arquivo armazena as mensagens rejeitadas em /export/home/perl/almost-certainly-spam e também no arquivo /export/home/perl/probably-spam.

Pode ser que alguma mensagem classificada como spam não seja realmente um spam. Isto é conhecido como "false positive". Neste caso, é importante que se tenha uma política de recuperação da mensagem. Por exemplo, quando a mensagem for retida pelo spamassasin, uma mensagem deverá ser enviada ao remetente avisando do ocorrido. No exemplo citado, isto é feito pelo script /etc/procmail.d/recusaSpam.pl. Além disso, deve-se guardar a mensagem retida por alguns dias, pois em caso de uma mensagem legítima ser retida, é possível recuperá-la.

Um detalhe muito importante para aumentar a eficiência do spamassassin é o seu treinamento. Isto é feito com mensagens reconhecidas como spam. Basta ter um conjunto de 1.000 ou mais spams em um folder (quantidade recomendada pelo fabricante do *software*) e treinar o spamassassin através do comando *sa-learn*. Admitindo que se tenha um folder, cujo nome é spam, com 1.000 ou mais mensagens, o treinamento é feito da seguinte forma:

sa-learn --spam --mbox spam

Quando o spamassassin bloqueia um spam, é gerado um relatório, o qual tem o motivo do porquê foi barrado. Segue o cabeçalho, para exemplificar, de uma mensagem considerada spam. A pontuação mínima para ser considerado spam, neste caso, é 4,9. No exemplo a seguir, a mensagem teve 11,7 pontos. Assim, foi classificada como spam e foi barrada.

Received: from 200.0.70.32 ([218.28.13.138]) by cnptia.embrapa.br (8.12.10/8.12.1) with SMTP id i2M2UcXV000485; Sun, 21 Mar 2004 23:31:05 -0300 (EST)

Received: from [11.163.175.43] by 200.0.70.32 with ESMTP id 09251313 for

<sac@cnptia.embrapa.br>; Sun, 21 Mar 2004 21:31:06 -0500

Message-ID: <1iyv$t8$733i9-71gh1$$ns54940@64lw5h21>

From: "Agilidade Cerebral" <iaco222@yahoo.com.br>

Reply-To: "Agilidade Cerebral" <iaco222@yahoo.com.br>

To: sac@cnptia.embrapa.br

Subject: Memorizando e Lendo Facilmente mv

X-Spam-Status: Yes, **hits=11.7 required=4.9** tests=BR_ADJUST_2,BR_CLIQUE_AQUI, BR_CURSO_BODY,BR_PERDER_TEMPO,BR_SPAMMER_URI,BR_VISITE,HTML_20 _30, HTML_MESSAGE,HTML_TAG_EXISTS_TBODY,LINES_OF_YELLING, MAILTO_TO_SPAM_ADDR,MSGID_FROM_MTA_SHORT,TO_MALFORMED autolearn=no version=2.70-cvs-spambr_20030926a

X-Spam-Checker-Version: SpamAssassin 2.70-cvs-spambr_20030926a (1.218-2003-11-09-exp) on norma.cnptia.embrapa.br

X-Spam-Flag: YES

X-Spam-Level: **********

MIME-Version: 1.0

Content-Type: multipart/mixed; boundary="----------=_405E509F.4485E008"

Content analysis details: (11.7 points, 4.9 required)

pts	rule name	description
0.3	TO_MALFORMED	To: has a malformed address
0.1	BR_CURSO_BODY	BODY: Curso no body
0.3	BR_VISITE	BODY: Fala sobre 'Visite nosso site'
0.3	BR_PERDER_TEMPO	BODY: Fala sobre nao perder tempo
0.0	HTML_MESSAGE	BODY: HTML included in message
0.1	HTML_TAG_EXISTS_TBODY	BODY: HTML has "tbody" tag
0.5	HTML_20_30	BODY: Message is 20% to 30% HTML
0.0	LINES_OF_YELLING	BODY: A WHOLE LINE OF YELLING DETECTED
2.0	BR_SPAMMER_URI	URI: Texto suspeito
1.1	MAILTO_TO_SPAM_ADDR	URI: Includes a link to a likely spammer email
1.8	BR_CLIQUE_AQUI	BODY: Contem o texto 'Clique aqui'
3.3	MSGID_FROM_MTA_SHORT	Message-Id was added by a relay
2.0	BR_ADJUST_2	Fortes caracteristicas +2

Uma ressalva sobre o spamassassin deve ser feita. Este *software,* quando da execução, consome muito recurso de CPU e memória. Assim, o servidor de correio eletrônico deverá ser uma estação com uma boa quantidade de memória RAM (pelo menos 256 MB) e CPU com clock acima de 400 MHz (arquitetura RISC). Se a arquitetura for CISC, recomenda-se memória RAM igual ou superior a 512 MB e CPU com clock acima de 1 Ghz.

Uma sugestão para diminuir o impacto do spamassassin no servidor, em relação ao consumo de CPU e memória, é a inserção de filtros no procmail que entreguem mensagens vindas do próprio domínio antes da mensagem ser avaliada pelo Spamassassim, por exemplo, além de outras situações já descritas anteriormente. Como exemplo, seria assim a configuração do procmailrc para que se evite ao máximo as chamadas para o software antispam (spamassassin ou outro qualquer):

1. verificar se a mensagem tem arquivo anexo com extensão não permitida (que pode ser vírus). Se tiver, rejeitar a mensagem e avisar ao remetente;
2. verificar se o remetente que tiver o domínio local existe. Se não existe, descartar. Caso contrário, entregar;
3. verificar se a mensagem vem de *sites* confiáveis. Se for verdade, então entregar;
4. inserir os filtros por palavras-chave ou frases, tal como já descrito;
5. analisar mensagem com filtro spamassassin ou outro filtro antispam.

Os passos de 1 a 4 são rapidamente executados pelo procmail, porém, o *script* do spamassassin, feito em perl, é mais demorado em relação aos passos de 1 a 4. Porém, uma grande quantidade de mensagens é entregue sem ser necessária a análise do spamassassin, visto que são rapidamente analisados nos passos de 1 a 4. Isto faz com que somente uma parcela do total de mensagens seja analisada pelo spamassassin, de forma que o servidor de correio eletrônico não fique sobrecarregado. Em suma, o desempenho do servidor melhora consideravelmente.

Bogofilter

Outra alternativa para barrar spams é o *software* bogofilter (Bogofilter, 2004). O Bogofilter é um filtro estatístico que constrói um dicionário de palavras que aparecem em um spam. Assim, este antispam usa o dicionário para determinar a probabilidade que uma dada mensagem seja um spam baseado nas palavras que este contém. O bogofilter é invocado a partir do procmail, tal como é feito com o spamassassin. A instalação e configuração do bogofilter pode ser vista no *site* do bogofilter (Bogofilter, 2004).

Para que o procmail venha a chamar o bogofilter, basta inserir as seguintes linhas no arquivo procmailrc, conforme (Dicas-L, 2004):

```
:0HB:

* ? bogofilter -u
/export/home/spam
```

Neste exemplo, caso a mensagem seja considerada spam, é movida para o diretório /export/home/spam da estação em questão.

Após instalar o bogofilter, é preciso treiná-lo. O treinamento é feito passando spams para o programa e, com a prática, ele vai aprendendo a diferenciar mensagens legítimas de spams. Para treinar o bogofilter, basta executar o comando

```
bogofilter -s < spam.archive
```

Neste comando, spam.archive é um folder que contém mensagens reconhecidas como spams. O ideal é ter um bom número deste tipo de mensagens (em torno de 1.000, pelo menos).

Para evitar erros, devido ao fato de mensagens legítimas serem confundidas com spam, as mensagens que o bogofilter classificar como spam e não o forem deverão ser armazenadas em um folder (notspam, por exemplo) e o bogofilter devera ser instruído (treinado) de que estas mensagens não são spams. Basta executar o comando

```
bogofilter -n < notspam
```

Em relação a consumo de recursos do servidor, as mesmas considerações sobre CPU e memória, feitas para o spamassassin, valem para o bogofilter.

O bogofilter pode ser inserido em procmailrc em conjunto com o spamassassin. Basta inserir a regra para invocá-lo, descrita anteriomente, após as regras do spamassassin. Assim, se a mensagem não for retida pelo spamassassin e for um spam, poderá ser retido pelo bogofilter.

Em Dicas-l (2004) existe uma relato sobre o bogofilter, no sentido de que seu uso teve uma eficiência de quase 100% de acertos. Tal como o spamassassin, o índice de acertos depende dos arquivos de configuração e também do treinamento. Mais detalhes, ver em Bogofilter (2004).

Tagged Message Delivery Agent -TMDA

Existe uma outra categoria de softwares antispams que agem de forma a certificar que o remetente é válido e, desta forma, verificar se a mensagem é de origem confiável. Para isto, este sistema, ao receber uma mensagem, envia uma outra mensagem para o remetente com a finalidade de desafiar o remetente a responder a uma pergunta ou tomar uma outra ação que seja fácil ao remetente. Caso o remetente responda corretamente, a mensagem original é entregue ao destinatário final. A seqüência seria:

1. o remetente envia mensagem para o destinatário;
2. o servidor de correio eletrônico retém a mensagem e envia uma outra ao remetente desafiando-o a responder alguma pergunta ou similar;
3. o remetente responde e a mensagem que contém a resposta vai para o servidor de correio eletrônico;
4. se a resposta ou ação tomada pelo remetente for correta, a mensagem original é entregue ao destinatário. Se a resposta estiver errada, a mensagem é descartada.

O TMDA (TMDA, 2004) é um *software* da categoria desafio/resposta. O desafio que o servidor envia ao remetente é que este responda à mensagem (reply to) simplesmente. Assim, o programa que envia spam provavelmente não irá responder à pergunta. A vantagem do TMDA é que ele é *freeware*, e também contém um sistema de lista branca ou negra. Assim, se o remetente estiver na lista branca do TMDA, nenhum desafio é feito. Se o remetente estiver na lista negra, a mensagem é descartada. Se o remetente não estiver em nenhuma destas listas, aí o servidor de correio eletrônico, usando o TMDA, envia uma mensagem desafiando o remetente a responder.

O TMDA tem a vantagem de evitar retenção de mensagens legítimas, o false positive. Porém, a entrega de mensagens, quando o remetente não está na lista branca, demora alguns minutos, desde que o remetente responda o desafio. Assim, é importante que o usuário mantenha a lista branca o mais atualizado possível. Para atualizar as listas branca ou negra, e também para atualizar ou inserir mais parâmetros de configurações, existe o tmda-cgi, um *software* que, após instalado, permite o usuário acessá-lo pela *web*. Mais detalhes pode ser visto em TMDA (2004).

O TMDA pode ser usado em conjunto com o spamassassin e demais filtros. A configuração é no arquivo procmailrc. Um exemplo é o que vai abaixo, supondo que o usuário que usar o TMDA seja ingres.

```
:0 c
* LOGNAME ?? ingres
  {

#######################################################
#
#                       Configuracao do TMDA
#
#######################################################

#
# Variaveis de ambiente
#
  LOGFILE=/var/log/procmail.log
  VERBOSE=yes
  LOGABSTRACT=all

# Set the necessary environment variables.
EXTENSION="$1"
:0
* EXTENSION ?? .
    {
        DELIMITER="+"
    }

RECIPIENT="$LOGNAME$DELIMITER$EXTENSION@cnptia.embrapa.br"
SENDER=`/usr/bin/formail -rtzxTo: | sed 's/[<>]//g;s/^[ ]*//' `
RETURN_PATH=$SENDER


# Run the message through tmda-filter.
#
:0 w
| /export/home/tmda/bin/tmda-filter

# Take the exit code from TMDA.
EXITCODE=$?

# TMDA takes care of final delivery
DEFAULT=/dev/null

}
```

Esta configuração foi feita para o usuário ingres. Como alguns usuários podem querer ou não usar o TMDA, deve-se inserir a regra acima para cada usuário que quiser o TMDA. Fica mais fácil se todos quiserem, pois aí é uma regra só para todos. Caso contrário, se terá n usuários que querem o TMDA,

assim é necessário ter n regras, uma para cada usuário.

Um detalhe interessante sobre o TMDA é que, durante os testes, feitos por 2 meses com alguns usuários, poucos foram os spams que passaram.O usuário ingres, por exemplo, teve 5 spams apenas. Os remetentes que enviaram spams tinham mecanismos automáticos de reply.

O TMDA tem um mecanismo automático de lista branca, isto é, se um remetente executar um reply, seu endereço vai automaticamente para a lista branca. O ideal é não configurar uma lista branca automática no TMDA pois, caso contrário, toda vez que um spam passar, o remetente é enviado para a lista branca e assim terá suas mensagens enviadas sem restrições. Este remetente pode ser um spammer.

Conclusões

Para se bloquear spams deve ser levado em conta um conjunto de soluções. Do que foi apresentado, inicialmente configura-se listas negras no arquivo de configuração do MTA. Em seguida, configura-se o arquivo procmailrc para que exista restrições quanto a extensões de arquivos anexos, usuários falsos, listas por palavras chaves ou frases e finalmente, o antispam, que deve ser o último a ser invocado pelo servidor para que o processamento não sobrecarregue o servidor.

Um detalhe sobre consultas a *sites* de lista negra é que o processamento pode ficar muito lento devido ao fato de cada mensagem requerer uma consulta. Assim, o ideal é evitar *sites* de lista negra o máximo possível, configurando o servidor para acessá-la em horários de pouco tráfego de mensagens (das 20:00h até 08:00h do outro dia, por exemplo).

A respeito das listas negras internas ao servidor (access), estas são extremamente úteis e retém uma considerável quantidade de spams, conforme visto na Tabela 1. Elas devem ser usadas sempre que possível, sem restrições.

As mensagens que passam pelas listas negras (interna e *sites*) são então enviadas para serem entregues. Neste ponto, os filtros para lista branca e negra configurados no arquivo procmailrc, descritos anteriormente, podem diminuir o processamento do servidor relativos aos processos dos softwares antispam. Como grande parte das mensagens são do próprio *site* local (remetente e destinatário), é possível verificar se o remetente pertence ao *site* local e, se for, entregar ao destinatário do *site* local. Se não for, descartar a mensagem. Além disso, *sites* confiáveis e lista branca podem ser configurados de forma que as mensagens provenientes destes possam ser entregues sem restrições. Por outro lado, os filtros de lista negra feito por palavras do tipo viagra, xanax, etc. ou por frases (click here, order here, etc.) são úteis para barrar spams que os *softwares* antispam do tipo spamassassin ou bogofilter não conseguem barrar. Além disso, os spams retidos por estes filtros podem ser usados para treinar os *softwares* antispam citados.

Enfim, além de contribuirem para diminuir o processamento relativo aos *softwares* antispam, os filtros configurados no arquivo procmailrc podem ser úteis para reter mensagens que serão usados para treinar o próprio software antispam.

A respeito dos *softwares* antispam, considerem-se o spamassassin e o TMDA. O spamassassin tem a vantagem de não ficar aumentando o tráfego na rede com as mensagens de desafio, mas tem uma grande desvantagem, isto é, pode reter mensagens legítimas, fenômeno conhecido por false positive. Para evitar danos por causa disto, deve ser feito um mecanismo que envia mensagens para o destinatário de forma que este, se este for legítimo, saiba que o sua mensagem foi retida e procure outra forma de enviar a mensagem. Outra desvantagem seria o fato de que não se consegue reter 100% de mensagens, mas consegue-se uma grande redução no número de spams. Outro detalhe importante sobre o spamassassin a ser mencionado é o fato de que, como os tipos de spams estão sempre mudando, treinamentos devem ser feitos sempre para que este tipo de antispam possa se adaptar às mudanças.

A respeito do TMDA, nos testes que foram feitos, pode-se dizer que ele reteve mais spams do que o spamassassin. Porém, *sites* que enviavam spams e que tinham mecanismos para dar reply aos desafios do TMDA conseguiram enviar spams. Desta forma, este *software* também não consegue reter 100% dos spams, mas mostrou ser mais eficiente do que o spamassassin, pelo menos nos testes realizados.

Para se usar o TMDA ou o Spamassassin, é muito importante que estes sejam chamados pelo mecanismos do servidor somente se forem realmente necessários pois consomem um razoável processamento por parte do servidor, o que poderá ser sentido principalmente se o *site* tiver um número elevados de usuários.

É possível atribuir um ou outro *software* antispam para cada usuário, ou até mesmo os dois, para se reter mais spams, mas isto terá um custo de processamento maior.

Finalizando, os spams podem ser barrados de diversas formas, desde mecanismos do próprio MTA (listas negras internas e acesso a sites de lista negra) até *softwares* específicos para este fim como o TMDA ou spamassassin.

Referências Bibliográficas

BOGOFILTER. Disponível em: <http://bogofilter.sourceforge.net>. Acesso em: 15 out. 2004.

COSTALES, B.; ALLMAN, E. **Sendmail**. Cambridge, Estados Unidos: O´Reilly, 1997.1021 p.

DICAS-L - **Lista dicas-l**. Disponível em: <http.www.dicas-l.unicamp.br>. Acesso em: 10 out. 2004.

NARCISO, M. G . **Instalação de antivírus na servidora de mail**: uma opção para impedir ataques de virus anexados a e-mail. Campinas: Embrapa Informática Agropecuária, 2001. 6 p. (Embrapa Informática Agropecuária. Instruções Técnicas, 4). Disponível em: <http://www.cnptia.embrapa.br/publica/2001/INSTR%20TECNICAS%204%20int.pdf>. Acesso em: 10 nov. 2004.

PERL - **Linguagem de Programação para CGI e scripts de propósitos gerais**. Disponível em: <http:www.perl.org>. Acesso em: 28 out. 2004.

POSTFIX . Disponível em: < http://www.postfix.org>. Acesso em: 28 out. 2004.

PROCMAIL. **Grupo de configuração do Procmail**. Disponível em: <http://www.ppgia.pucpr.br/~borchardt/tools/>. Acesso em: 28 out. 2004.

QMAIL . Disponível em: <http://www.qmail.org>. Acesso em: 28 out. 2004.

SCANMAILS. Disponível em: <http://www.scan.org>. Acesso em: 28 out. 2004.

SENDMAIL. Disponível em: <http://www.sendmail.org>. Acesso em: 15 mar. 2004.

SPAMASSASSIN. Disponível em: <http://www.spamassassin.org>. Acesso em: 15 out. 2004.

TMDA. **Tagged Message Delivery Agent (TMDA)**. Disponível em: <http://www.tmda.net>. Acesso em: 15 out. 2004.

Ministério da Agricultura,
Pecuária e Abastecimento

BRASIL
UM PAÍS DE TODOS
GOVERNO FEDERAL