

Configurando um servidor de e-mails

Tradicionalmente, o **Sendmail** é o servidor de e-mails mais conhecido, não apenas no Linux, mas nos sistemas Unix em geral. Ele é um dos mais antigos (disponível desde 1982, mais de uma década antes da popularização da Internet) e foi a opção padrão de 9 em cada 10 administradores de sistemas durante muito tempo.

Apesar disso, o uso do Sendmail vem decaindo de forma estável de uma década para cá. As queixas podem ser resumidas a duas questões fundamentais. A primeira é o brutal número de opções e recursos disponíveis, que tornam a configuração bastante complexa e trabalhosa. Muitos administradores da velha guarda gostam da complexidade, mas a menos que você pretenda dedicar sua via à arte de manter servidores Sendmail, ela acaba sendo um grande problema.

A segunda questão é o histórico de vulnerabilidades do Sendmail que, na melhor das hipóteses, pode ser definido como "muito ruim". É verdade que nos últimos anos as coisas melhoraram bastante, mas as cicatrizes do passado ainda incomodam.

O concorrente mais antigo do Sendmail é o **Exim**, que oferece um conjunto bastante equilibrado de recursos, boa performance e um bom histórico de segurança. O EXIM é o MTA usado por padrão no Debian, ele é instalado automaticamente como dependência ao instalar pacotes que necessitem de um servidor de e-mails, mas pode ser rapidamente substituído pelo Postfix ou o Sendmail via apt-get, caso desejado.

O **Qmail** é uma escolha mais complicada. Quando foi lançado, em 1997, o Qmail trouxe várias inovações e um design bastante simples e limpo, com ênfase na segurança, o que o tornou rapidamente uma opção bastante popular.

Entretanto, o Qmail possui dois graves problemas. Ele foi abandonado pelo autor em 1998, depois do lançamento da versão 1.03 e, embora o código fonte seja aberto, a licença de uso impede a redistribuição de versões modificadas, embora seja permitido disponibilizar patches.

Ao longo dos anos, surgiram várias iniciativas de atualizações do Qmail, onde o código original é distribuído junto com um conjunto de patches com atualizações. Para instalar, você precisa primeiro aplicar cada um dos patches, para em seguida poder compilar e instalar o Qmail. Dois dos projetos mais populares são o <http://qmail.org/netqmail/> e o <http://www.qmailrocks.org/>.

Embora o Qmail ainda possua uma legião de seguidores fiéis, a limitação imposta pela licença acaba sendo um grande empecilho para quem deseja utilizá-lo e representa uma grande ameaça à manteneabilidade dos patches a longo prazo, já que as alterações em relação ao código original tornam-se cada vez mais complexas e difíceis de aplicar, com a disponibilização de patches para patches que já são patches para outros patches.. ;).

Finalmente, temos o **Postfix**. Ele é uma espécie de meio termo entre a simplicidade do Qmail e a fartura de recursos do Exim. Entre os três, ele é o mais rápido e o mais simples de configurar, o que faz com que ele seja atualmente o mais popular e o que possui mais documentação disponível. O Postfix também possui um excelente histórico de segurança, sendo considerado por muitos tão seguro quanto o Qmail.

Existem fortes motivos para não usar o Sendmail ou o Qmail em novas instalações, mas temos uma boa briga entre o Postfix e o Exim. Escolhi abordar o Postfix aqui simplesmente por que, entre os dois, ele é mais popular, o que torna mais simples encontrar documentação e conseguir ajuda quando tiver dúvidas.

Apesar disso, a maior parte dos conceitos podem ser usados também na configuração do Sendmail e outros servidores; afinal, a configuração de todos eles reserva mais semelhanças que diferenças.

Instalando o Postfix

O pacote do Postfix pode ser encontrado em todas as principais distribuições. Nas distribuições derivadas do Debian, você pode instalá-lo usando o apt-get:

```
# apt-get install postfix
```

Mais três pacotes que adicionam algumas funcionalidades importantes são:

```
# apt-get install postfix-ldap
```

(permite configurar o servidor para obter a lista de logins e senhas a partir de um servidor LDAP)

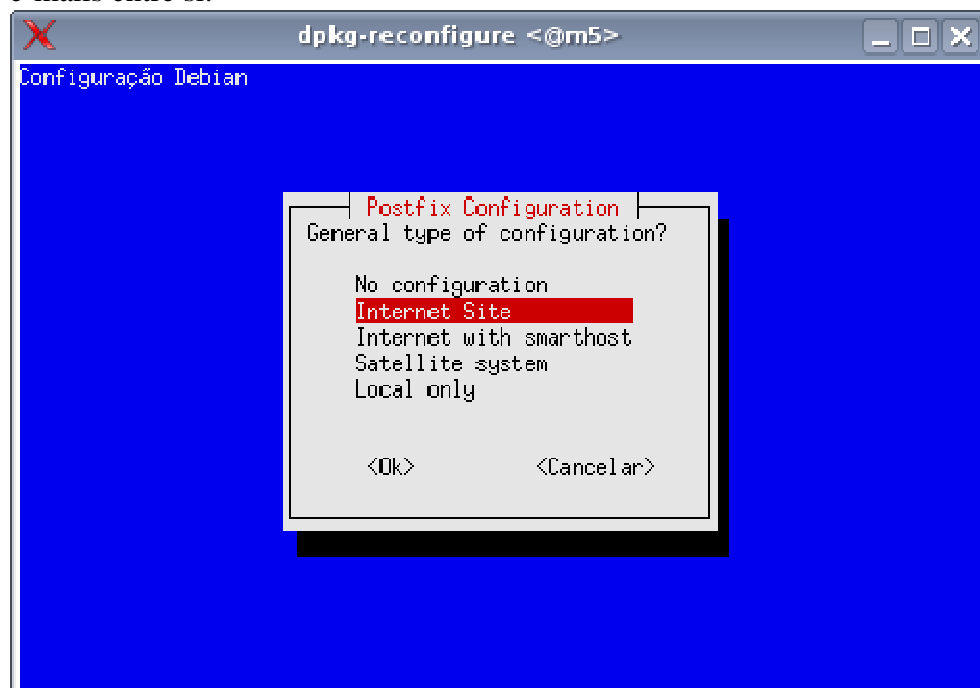
```
# apt-get install postfix-mysql
```

```
# apt-get install postfix-pgsql
```

(para usar um servidor MySQL ou Postgree para armazenar a lista de logins e senhas)

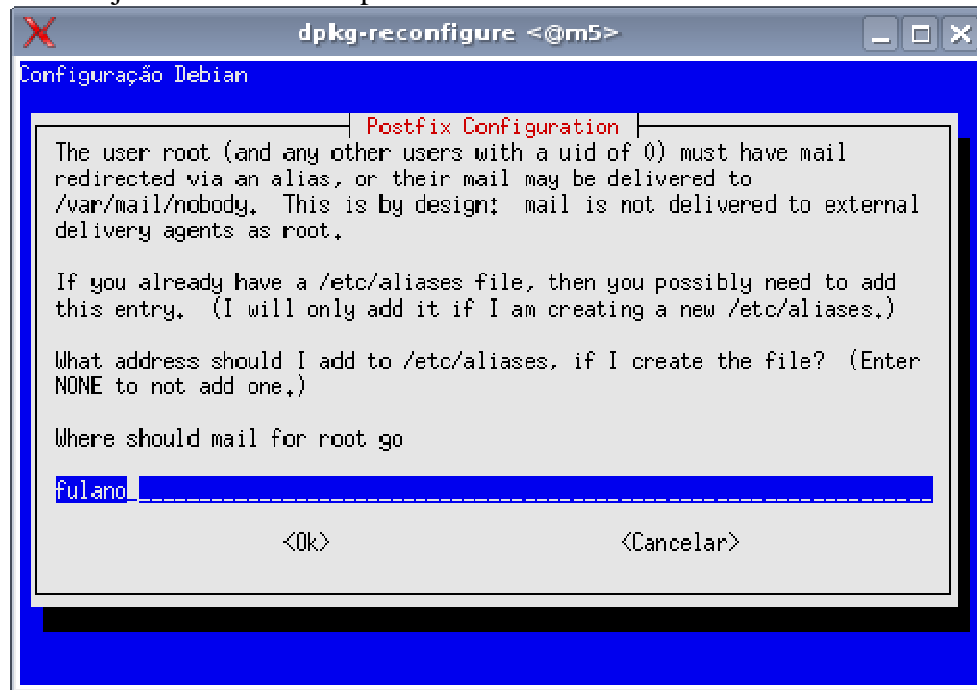
O pacote do Debian possui um wizard configuração, exibido durante a instalação do pacote. Ele faz algumas perguntas e se encarrega de gerar uma configuração básica, suficiente para colocar o servidor para funcionar. Ele não faz nada de sobrenatural, apenas ajusta o "/etc/postfix/main.cf" de acordo com as opções definidas. Por enquanto, vou apenas explicar rapidamente as opções, pois as veremos com mais detalhes ao estudar a configuração manual do postfix.

A primeira pergunta é sobre a função do servidor de e-mails que você está configurando. A opção mais usada é "Internet Site", onde você cria um servidor "de verdade", que envia e recebe os e-mails diretamente. Na segunda opção "with smarthost" seu servidor recebe mensagens, mas o envio fica a cargo de outra máquina, enquanto na terceira ("Satellite system", a mais limitada) seu servidor envia através de outra máquina e não recebe mensagens. A opção "Local only" é usada apenas em redes de terminais leves (poderia ter alguma utilidade num servidor LTSP, por exemplo), pois permite apenas que os usuários logados no servidor troquem e-mails entre si.

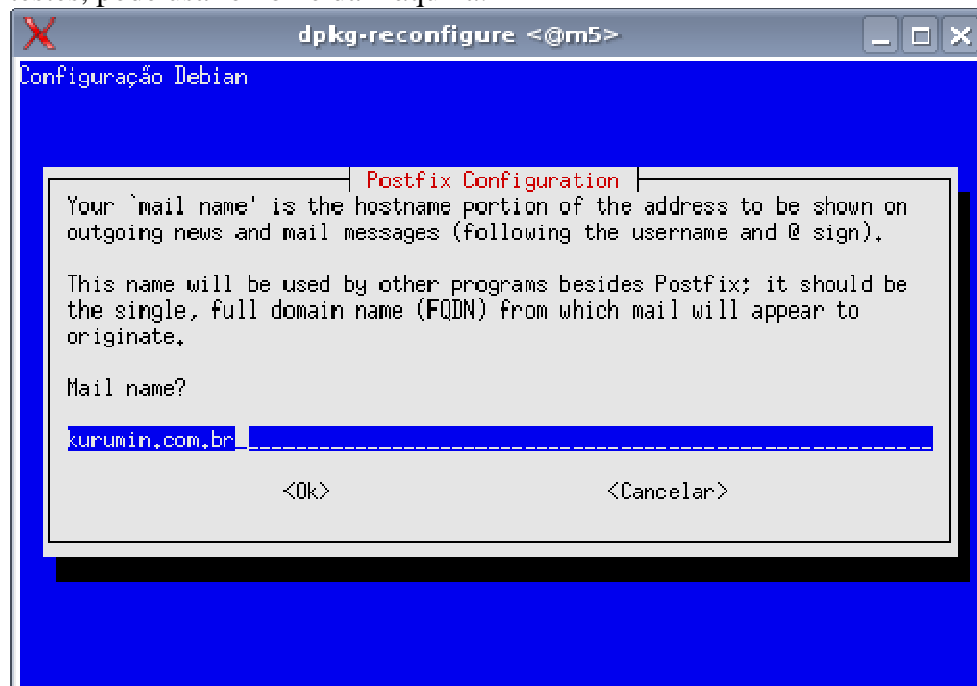


Nos sistemas Linux, é recomendado que você use a conta root apenas para a manutenção do sistema. Mesmo sendo o administrador, você usa uma conta normal de usuário, utilizando o su ou sudo para ganhar privilégios

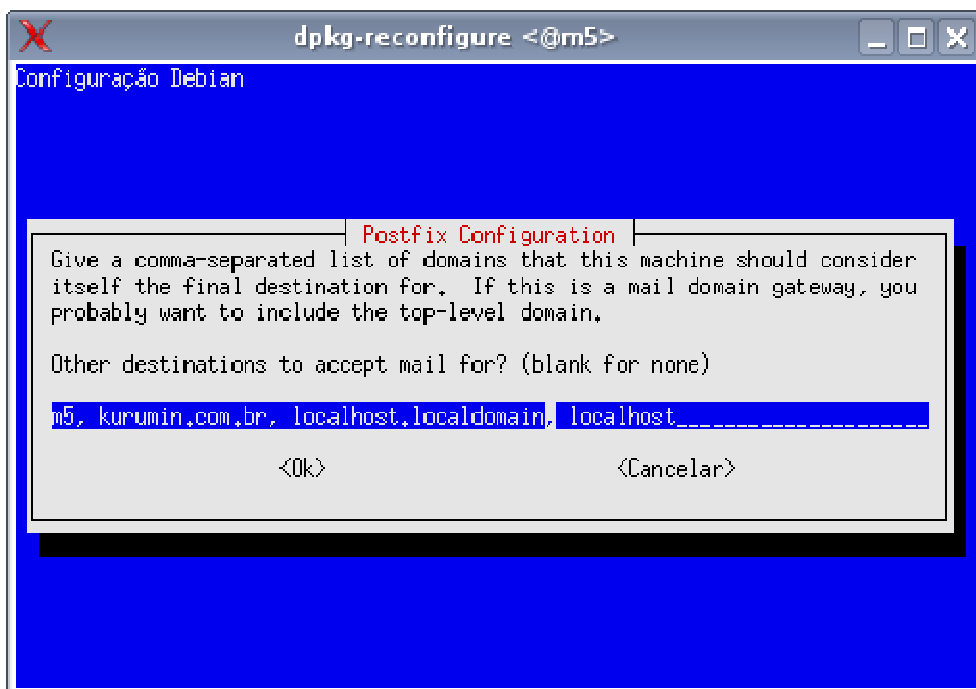
de root apenas quando necessário. Se você quase nunca usa a conta root, significaria que os e-mails enviados para o "root@seu-servidor" nunca seriam lidos. A segunda pergunta mata a questão, permitindo que os e-mails sejam encaminhados para sua conta de usuário:



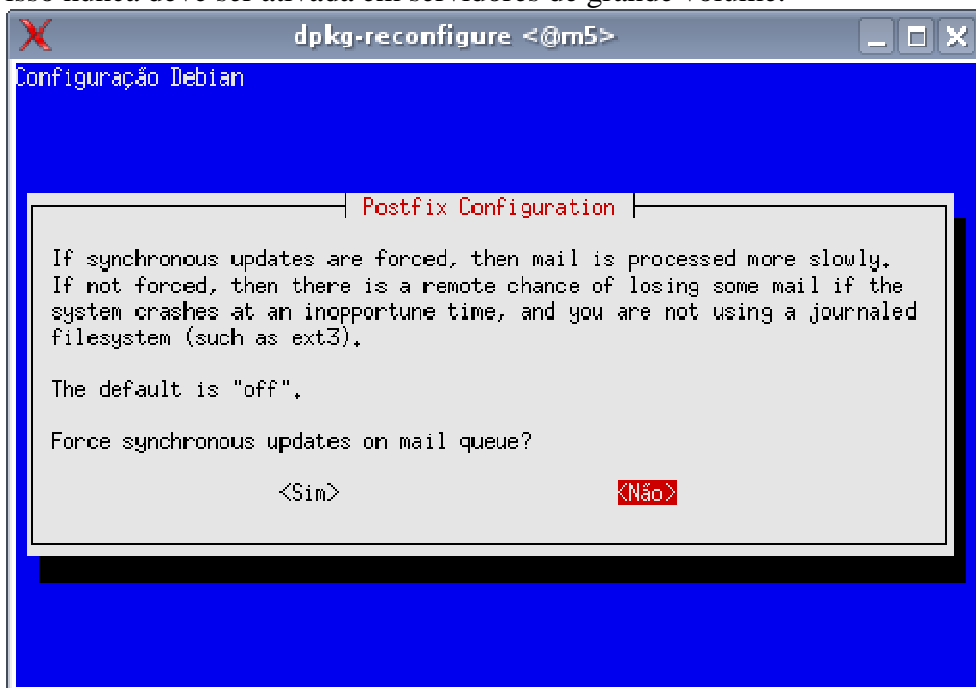
A terceira pergunta é sobre o domínio do servidor, que será incluído nas mensagens enviadas. Se o você está configurando um servidor dedicado use seu domínio registrado. Se está apenas configurando um servidor de testes, pode usar o nome da máquina:



A questão seguinte já é um pouco mais complexa. Você deve definir os destinos que serão aceitos pelo seu servidor, ou seja, os endereços que colocados no destinatário da mensagem fazem ele entender que o e-mail é para ele. Aqui você usa o nome da máquina, o domínio registrado (no caso de um servidor real), "localhost.localdomain" e "localhost", todos separados por vírgula e espaço. esta forma, qualquer e-mail destinado ao "fulano@kurumin.com.br", "fulano@m5" (o nome da máquina) ou "fulano@localhost", que chegue até seu servidor, será encaminhado para a caixa postal do usuário "fulano". Em compensação, um e-mail destinado ao "ciclano@guiadohardware.net", por exemplo, será repassado ao servidor responsável pelo domínio correspondente.



A opção "synchronous updates" permite desativar as otimizações no envio das mensagens, fazendo com que os e-mails sejam enviados conforme são recebidos e em ordem. Esta opção aumenta um pouco a confiabilidade do servidor, pois reduz a possibilidade de perda de mensagens ainda não enviadas, em casos de travamentos ou quedas de energia. Por outro lado, ela reduz substancialmente o desempenho do servidor, por isso nunca deve ser ativada em servidores de grande volume.



Depois de concluída a instalação, o servidor já estará iniciado e configurado para subir automaticamente durante o boot. Em algumas distribuições, como no Mandriva, o servidor é configurado para subir durante o boot, mas não fica ativado depois da instalação, para que você tenha a chance de revisar o arquivo de configuração antes de ativá-lo. Neste caso, você precisa iniciar o servidor manualmente usando o comando "service postfix start", ou "/etc/init.d postfix start".

O servidor SMTP escuta, por padrão, na porta 25. Os e-mails são transmitidos de uma forma bem simples, com comandos de texto. Uma forma de entender como isso funciona é mandar um e-mail interno para o root do sistema, usando o telnet.

Sim, os servidores SMTP podem ser acessados via telnet, basta mandar o cliente se conectar na porta 25. Isso permitirá enviar o e-mail de testes conversando direto com o servidor Postfix. Se o IP do servidor na rede interna for 192.168.1.33, por exemplo, o comando seria:

```
$ telnet 192.168.1.33 25
```

```
Trying 192.168.1.33...
Connected to 192.168.1.33.
Escape character is '^]'.
220 kurumin ESMTP Postfix (Debian/GNU)
HELO smtp.eu.com
250 kurumin
MAIL From: eu@eu-mesmo.com
250 Ok
RCPT to: joao@localhost
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Vai ver se estou na esquina!
.
250 Ok: queued as 8CEDB2215
QUIT
221 Bye
Connection closed by foreign host.
```

As linhas em negrito são os comandos executados no terminal, seguidos pelas respostas do servidor. O comando "HELO" serve para iniciar a conversa, onde o emissor se identifica. Os passos seguintes são dizer o emissor do e-mail (MAIL From:) e o destinatário (RCPT to:), seguido pelo texto do e-mail (DATA). Note que depois do texto vem uma linha com um ponto, que indica o final da mensagem.

No caso, enviei um mail com remetente falso para o usuário "joao" da máquina (joao@localhost). Este e-mail local pode ser lido usando um cliente de modo texto, como o mutt:

```
# apt-get install mutt
```

Da próxima vez que você se logar com o usuário especificado, verá uma mensagem avisando da polida mensagem que foi enviada:

You have new mail.

Chamando o mutt, você verá que e-mail realmente está lá.

Antigamente, antes da popularização da internet, esses e-mails locais eram comuns, pois geralmente várias pessoas usavam o mesmo servidor (e cada servidor possuía vários terminais burros ligados a ele). As mensagens eram trocadas diretamente entre os servidores e armazenadas no spool. Quando o usuário se logava, tinha acesso à sua caixa postal.

Hoje em dia, pouca gente ainda utiliza o mutt. Em geral usamos servidores POP3 ou IMAP para armazenar as mensagens e as baixamos de vez em quando usando algum cliente de e-mails gráfico. A idéia continua sendo basicamente a mesma, mas agora em escala muito maior. Cada e-mail enviado passa por vários servidores antes de chegar ao destinatário, as mensagens são armazenadas no servidor POP3 ou IMAP do servidor e, quando o destinatário se conecta, baixa todas as mensagens de uma vez.

O Postfix (ou Qmail ou Sendmail) armazena as mensagens em uma pasta local, por padrão a pasta "Maildir", dentro do diretório home de cada usuário. Programas como o Mutt acessam diretamente as mensagens dentro da pasta, mas, para baixar as mensagens remotamente, via pop3 ou imap, você precisa instalar um componente adicional.

Existem vários servidores pop3, como o Cyrus e o Courier. O Courier é o mais usado, pois inclui vários componentes adicionais. Ele é, na verdade, uma suíte, que inclui até mesmo um webmail.

Para instalar o módulo pop3, instale o pacote:

```
# apt-get install courier-pop
```

Aproveite para instalar também a encriptação via ssl. Este recurso é importante hoje em dia, pois sem encriptação, seus e-mails (incluindo o login e senha) são transmitidos em texto plano pela rede e podem ser interceptados. Uma vez ativo o recurso no servidor, basta marcar a opção no cliente de e-mails.

```
# apt-get install courier-pop-ssl
```

Para instalar o servidor imap, instale os pacotes:

```
# apt-get install courier-imap
```

```
# apt-get install courier-imap-ssl
```

Com esta configuração básica você já conseguirá enviar e receber e-mails. Inicialmente, você pode testar pedindo para alguém enviar um e-mail para seu endereço IP, como em: fulano@200.220.123.32. Se tudo estiver funcionando, o próximo passo é configurar o servidor DNS (<http://www.guiadohardware.net/tutoriais/120/>) para que você possa receber e-mails através do seu domínio registrado.

Não é uma boa idéia receber e-mails usando uma conexão ADSL, pois uma conexão instável fará com que alguns e-mails sejam perdidos. Outro problema é que quase todas as faixas de endereços de conexões via ADSL nacionais fazem parte de listas negras de spam (justamente por já terem sido exaustivamente usadas para envio de spam no passado). Nesse caso, é melhor configurar seu servidor como um sistema satélite, onde é usado um servidor SMTP externo para envio de e-mails. Você pode usar o próprio SMTP do provedor, ou o servidor de uma empresa de hospedagem, que tenha o nome "limpo" na praça.

De qualquer forma, nada impede que você registre uma conta em um serviço de DNS dinâmico, como o <http://no-ip.com> e experimente manter seu servidor de e-mails para fins didáticos.

Ao usar os pacotes **courier-pop-ssl** ou **courier-imap-ssl**, é necessário gerar um certificado. Empresas como a Verisign vendem certificados reconhecidos, que são necessários caso você queira abrir um site de comércio eletrônico, por exemplo. Mas, para um servidor particular, não existe nada errado em gerar seu próprio certificado. Ele vai funcionar da mesma forma e, se corretamente gerado, com a mesma segurança. O único efeito desagradável é que os clientes receberão uma mensagem "Não é possível comprovar a autenticidade do certificado..." ao se conectarem.

Para criar uma chave para o servidor **IMAP**, comece renomeando a chave de testes, criada durante a instalação:

```
# cd /etc/courier
```

```
# mv imapd.pem imapd.pem.old
```

Edite agora o arquivo "**imap.conf**" (na mesma pasta), colocando as informações sobre o país, cidade, nome do servidor, etc. Depois, basta gerar o novo certificado com o comando:

```
# mkimapdcert
```

Para gerar a chave para o servidor POP3, o procedimento é quase o mesmo. Dentro da pasta "/etc/courier" remova ou renomeie o arquivo "pop3d.pem", edite o arquivo "**pop3d.cnf**", colocando as informações do servidor e gere o certificado com o comando "**mkpop3dcert**".

Cadastrando usuários e Configurando

Lendo a documentação, parece que cadastrar usuários no servidor de e-mails é muito complicado, pois existem muitas formas de fazer isso. A forma mais simples é simplesmente criar um novo usuário no sistema, usando o comando adduser, como em:

```
# adduser joao
```

Desde que o servidor de e-mails esteja instalado, será criada a conta `joao@servidor`, acessível tanto localmente (usando o `mutt` e outros clientes), quanto remotamente, via `pop3` ou `imap`.

O problema é que o usuário `joão` passa a poder logar-se na máquina de outras formas, via `ssh`, `telnet`, acessar compartilhamentos de rede e assim por diante. Essa abordagem serve para servidores internos, onde os usuários são conhecidos ou funcionários da mesma empresa, mas não é um sistema adequado para um grande servidor web, com inúmeras contas de e-mails de usuários desconhecidos ou que hospeda um servidor Apache com vários subdomínios.

Hoje em dia existem várias outras opções para cadastrar contas no servidor de e-mails, sem precisar necessariamente criar logins válidos no sistema. Você pode armazenar as contas em um servidor MySQL, PostgreSQL ou até mesmo em um servidor LDAP. Para isso, usamos os pacotes **postfix-ldap**, **postfix-mysql** ou **postfix-pgsql**, que vimos anteriormente.

Configurando

Antes de começar a falar sobre a configuração do Postfix, é importante que você entenda alguns termos usados com frequência nos arquivos de configuração e tutoriais:

- **MTA** (Mail Transport Agent): É o servidor de e-mails propriamente dito, com o Postfix, Qmail, Sendmail e o Exim. Um MTA obrigatoriamente suporta enviar e receber e-mails via SMTP, o protocolo utilizado para transportar as mensagens entre os servidores. O servidor pode ser configurado para enviar e receber os e-mails diretamente (internet site) ou se limitar a receber mensagens, usando um servidor SMTP externo (smarthost) na hora de enviar.

Normalmente, você configura seu servidor como "internet site" apenas ao utilizar um servidor dedicado, ou caso sua empresa possua um link dedicado, com um IP "limpo", fora dos cadastros das listas negras de spam (você pode checar através do <http://rbls.org/>).

- **Mua** (Mail user agent): Este é o nome técnico do cliente de e-mail, como o Thunderbird, Evolution, Kmail, etc. usados diretamente pelo usuário final.
- **MDA** (Mail Delivery Agent): O MDA funciona como um intermediário entre o MTA e o Mua. Ele não é obrigatório, mas pode fazer algumas coisas úteis, como aplicar filtros antispam, remover vírus anexados nas mensagens, encaminhar para outros endereços e assim por diante. Dois exemplos usados no Linux são o Fetchmail e o Procmail. Você os utiliza quando precisa baixar as mensagens do provedor e aplicar filtros diversos antes de encaminhá-las aos usuários.

O principal arquivo de configuração do Postfix é o `"/etc/postfix/main.cf"`. Este é um exemplo de arquivo de configuração funcional. Veja que, apesar da complexidade da tarefa, a configuração do Postfix é relativamente simples:

```
# /etc/postfix/main.cf
myhostname = etch.kurumin.com.br
mydomain = kurumin.com.br
append_dot_mydomain = no
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = etch.kurumin.com.br, kurumin.com.br, localhost
```

```
relayhost =  
mynetworks = 127.0.0.0/8  
home_mailbox = Maildir/  
mailbox_command =  
recipient_delimiter = +  
inet_interfaces = all  
inet_protocols = all  
message_size_limit = 20000000  
mailbox_size_limit = 0
```

Vamos a uma explicação mais detalhada de cada uma das opções:

Embora não seja citado no arquivo, o postfix roda utilizando uma conta com privilégios limitados, de forma a limitar o dano no caso de qualquer problema de segurança relacionado ao servidor de e-mails). Estas opções já vem configuradas por padrão ao instalar o pacote.

As primeiras linhas do arquivo indicam o nome da máquina e o domínio. Caso seu servidor não tenha um domínio registrado, ou é usado apenas dentro da rede local), você pode usar o "localdomain" como domínio. Note que muitos servidores rejeitam mensagens enviadas por servidores sem domínio registrado, para dificultar o envio de spams. É por isso que é tão importante configurar corretamente o DNS reverso no Bind, já que é através dele que os servidores remotos podem verificar se os e-mails realmente vêm do seu domínio.

A opção "myhostname" deve conter o nome completo do servidor, incluindo o domínio, enquanto que a opção "mydomain" contém apenas o domínio, sem o nome da máquina, como em:

```
myhostname = etch.kurumin.com.br  
mydomain = kurumin.com.br  
append_dot_mydomain = no
```

A linha "mydestination" Esta linha indica quais nomes e domínios serão considerados endereços locais pelo servidor. Se o nome do servidor é "kurumin.kurumin.com.br" e o "domínio "kurumin.com.br", o servidor entenderia que tanto e-mails endereçados a "usuario@etch.kurumin.com.br", quanto "usuario@kurumin.com.br" e "usuario@localhost" são endereçados a ele mesmo.

```
mydestination = etch.kurumin.com.br, kurumin.com.br, localhost
```

A linha "mynetworks" especifica os endereços ou faixas de endereços a partir de onde o servidor aceitará o envio de mensagens.

É preciso configurar esta opção com muito cuidado, caso contrário um spammer poderá usar seu servidor para enviar mensagens não solicitadas, consumindo sua banda e possivelmente fazendo seu servidor ser incluído em várias blacklists, o que vai lhe causar muita dor de cabeça.

A opção 'mynetworks = 127.0.0.0/8' permite apenas e-mails enviados localmente. Você pode especificar várias faixas de endereços separando-os com vírgula, como em: "mynetworks = 200.221.149.0/24, 127.0.0.0/8".

```
mynetworks = 127.0.0.0/8  
inet_interfaces = all
```


Na opção "relayhost" você pode indicar um servidor SMTP externo, através do qual as mensagens serão enviadas. Deixando esta opção em branco, todos os e-mails serão enviados diretamente pelo seu servidor.

Hoje em dia é bem mais simples usar um servidor externo por causa da questão do spam. Usar o smtp de um provedor conhecido fará com que menos mensagens se percam nos filtros dos destinatários.

Para usar um relayhost aqui, é preciso indicar um servidor que aceite mensagens enviadas por este servidor sem pedir autenticação. Em geral, as empresas que oferecem serviços de hospedagem oferecem esta opção em troca de uma taxa adicional. É possível também configurar seu provedor para se autenticar, com um pouco mais de trabalho.

Ex: relayhost = smtp.meuprovedor.com

Opcionalmente, você pode configurar os clientes de e-mail nas estações para usarem diretamente o smtp do provedor, deixando seu servidor postfix apenas para receber. Nesse caso, você pode usar qualquer smtp a que tenha acesso.

relayhost =

A linha "home_mailbox" indica a pasta local, dentro do home de cada usuário, onde os e-mails ficarão armazenados. A pasta Maildir/ é o padrão usado por diversos MTA's. Caso necessário, crie a pasta manualmente, usando o comando "maildirmake ~/Maildir" (executado como o usuário para o qual a pasta será criada).

Em seguida, execute o comando "maildirmake /etc/skel/Maildir" como root, para que todos os novos usuários criados a partir daí já venham com a pasta criada. Normalmente, os pacotes instalados pelas distribuições automatizam esta etapa.

```
home_mailbox = Maildir/  
recipient_delimiter = +  
inet_interfaces = all
```

Na maioria dos casos, é desejável limitar o tamanho das mensagens recebidas, para evitar que algum espertinho envie um ISO de CD anexado à mensagem, consumindo toda a banda e acabando com o espaço em disco do servidor. O padrão do postfix é limitar as mensagens a 10 MB. Qualquer anexo maior que isso é recusado. Esta configuração pode ser alterada através da opção "message_size_limit", onde você especifica o valor desejado, em bytes. Note que por causa do uso do MIME, o tamanho dos anexos cresce substancialmente ao serem enviados via e-mail. Um arquivo de 5 MB, transforma-se numa mensagem de quase 7. Leve isto em conta ao definir o limite. Aqui estou usando um limite de 20 MB decimais:

```
message_size_limit = 20000000
```

A opção "mailbox_size_limit" serviria para definir o limite de armazenamento para a caixa postal do usuário. Entretanto, ao usar o formato Maildir para as caixas postais, cada mensagem é salva num arquivo separado, de forma que a opção não funciona. Por isso, usamos o valor "0" para desativá-la. A melhor forma de limitar o espaço dos usuários é simplesmente definir quotas de espaço em disco, usando o Quota.

```
mailbox_size_limit = 0
```

Em geral, os arquivos de configuração padrão, incluídos nas distribuições, são suficientes para ter um servidor de e-mails básico funcional. Mas, depois de feito o primeiro teste, nunca deixe de editar o arquivo, verificando todas as opções. Você pode tanto usar como ponto de partida o arquivo original, quanto usar este modelo.

Com o tempo, o ideal é que você desenvolva um arquivo próprio, com as opções que você usa mais freqüentemente e comentários que lhe ajudem a lembrar como e em quais situações usar cada uma. Lembre-se de que, salvo eventuais diferenças entre as versões instaladas, um arquivo de configuração usado no Fedora ou Mandriva vai funcionar perfeitamente no Debian, Slackware, ou em qualquer outra distribuição que siga um nível mínimo de padrões. O software em si, o Postfix, será o mesmo, independentemente da distribuição usada.

Instalando um webmail

O Squirrelmail é um script de webmail escrito em php, que permite acessar as mensagens de um servidor imap via web. Ele é bem leve, tanto do ponto de vista dos recursos utilizados no servidor, quanto do ponto de vista dos clientes. As páginas geradas pelo webmail são simples páginas html, sem javascript nem nenhum outro recurso especial. Isso o torna um campeão de compatibilidade, principalmente com os navegadores usados em PDAs e browsers antigos.

Para instalar o Squirrelmail você vai precisar do seguinte:

1- Um servidor Postfix (ou outro MTA suportado), com suporte a IMAP, o que inclui basicamente os pacotes "postfix" e "courier-imap". Siga as instruções anteriores para instalar o servidor de e-mails e criar as contas de usuários.

2- Um servidor Apache, com suporte a PHP4 instalado. Tanto faz usar o Apache 1.3 ou o Apache 2, o Squirrelmail roda em ambos, verifique apenas se o suporte a PHP está realmente instalado e funcionando.

Satisfeitos esses dois pré-requisitos, o Squirrelmail em si é bem simples de instalar. Você tem duas opções. Pode instalá-lo usando o gerenciador de pacotes da sua distribuição ou baixar o arquivo manualmente. A principal vantagem em usar o pacote incluído na distribuição é que a instalação é feita com checagem de dependências, o que é uma segurança a mais contra eventuais barbeiragens na configuração do Postfix ou do Apache.

No Debian, por exemplo, você pode instalá-lo usando o apt-get:

```
# apt-get install squirrelmail
```

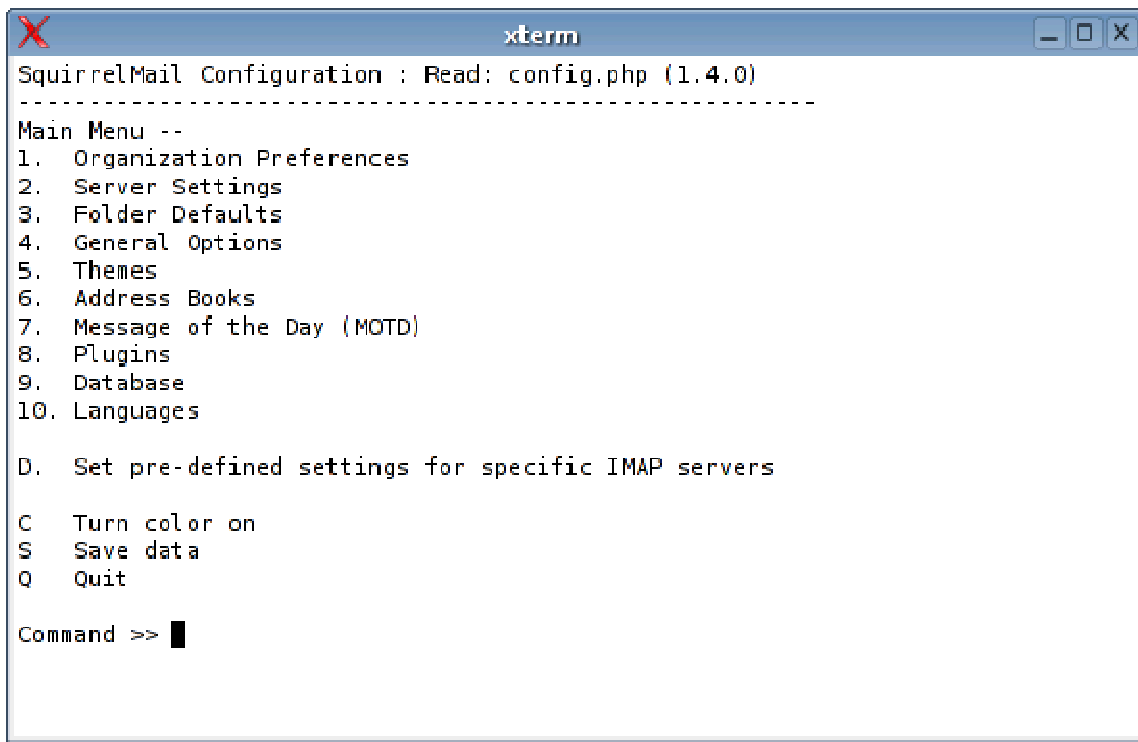
O Squirrelmail é instalado por padrão dentro da pasta `"/usr/share/squirrelmail"`, que fica fora da jurisdição do servidor web. Existem várias formas de fazer com que o webmail fique acessível apesar disso. Você pode, por exemplo, criar um link dentro da pasta `"/var/www/"` apontando para a pasta `"/usr/share/squirrelmail"`. Mas, uma forma mais elegante de ter o mesmo resultado, é adicionar as duas linhas abaixo no arquivo `"/etc/apache2/httpd.conf"`:

```
Alias /webmail "/usr/share/squirrelmail/"
DirectoryIndex index.php
```

Aqui estamos criando uma pasta virtual "webmail/" no servidor web, que aponta para o arquivo index.php dentro da pasta real.

Ao baixar manualmente, pegue o arquivo no <http://www.squirrelmail.org/download.php> e copie o conteúdo do arquivo para uma pasta dentro do seu servidor web, como, por exemplo, `"/var/www/webmail"`; basta descompactar o arquivo, como no caso do phpBB. Não é necessário compilar nada. Opcionalmente, você pode usar a pasta `"/usr/share/squirrelmail"` e adicionar a entrada do alias no arquivo de configuração do Apache.

Depois de instalar, é preciso fazer a configuração básica do Squirrelmail, usando o utilitário **"squirrelmail-config"**. Se você instalou a partir do pacote, pode chamá-lo diretamente (como root) a partir do terminal. Se tiver instalado manualmente, execute o script **"configure"**, dentro da pasta do Squirrelmail.



```
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color on
S Save data
Q Quit

Command >> █
```

A configuração mínima inclui:

- a) Defina o nome da empresa, logotipo, URL, etc. na opção **1**.
- b) Defina o domínio do servidor (ex: minhaempresa.com) na opção **2**. Se você está configurando um servidor local, sem usar um domínio registrado, mantenha o default.
- c) Ainda na opção **2**, verifique se as configurações de servidor estão corretas. Elas devem ser:

3. Sendmail or SMTP : SMTP

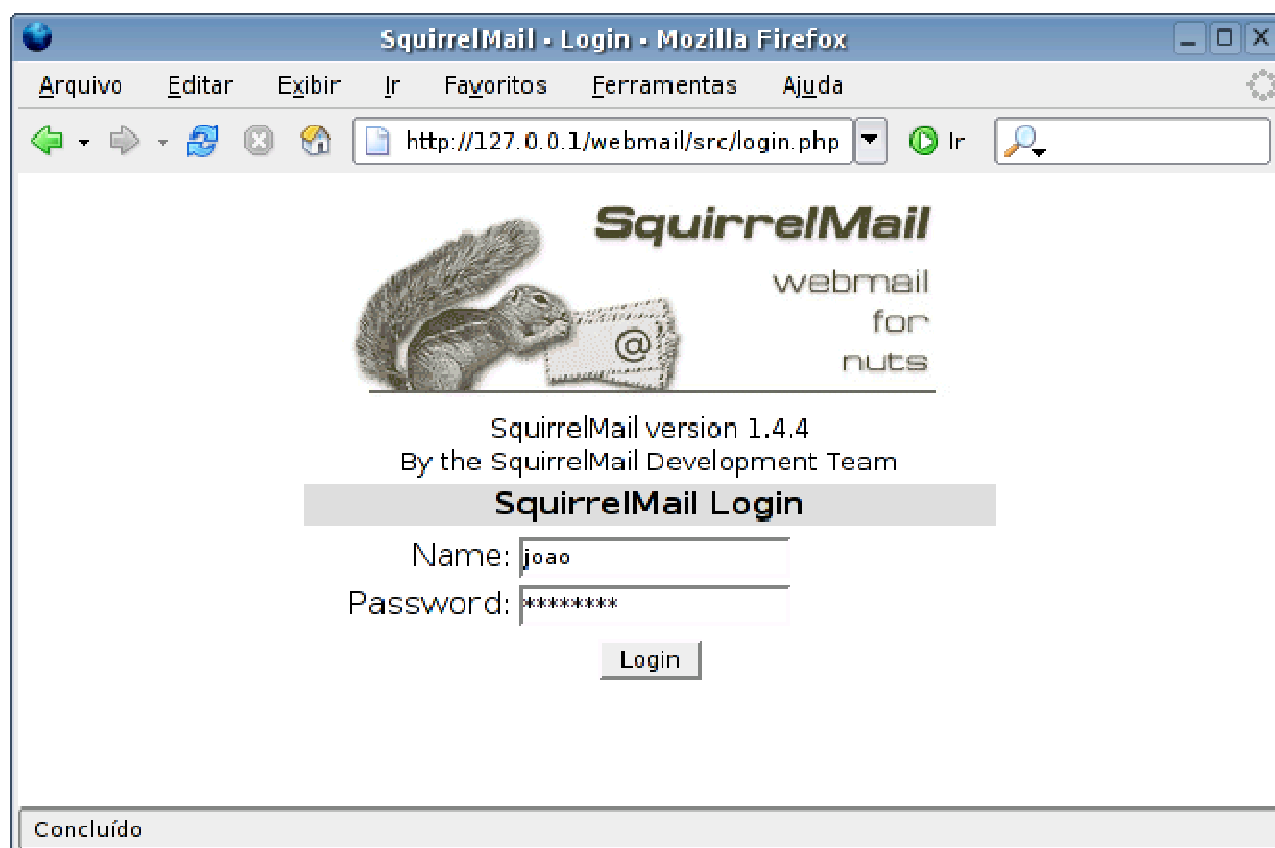
A. Update IMAP Settings : localhost:143 (other)

B. Update SMTP Settings : localhost:25

O Squirrelmail pode ser configurado para utilizar outros servidores. Você pode usar o Sendmail (ou Qmail) no lugar do Postfix ou utilizar outros servidores IMAP além do Courier. As configurações acima são as que se aplicam no nosso caso, usando o postfix e o courier-imap.

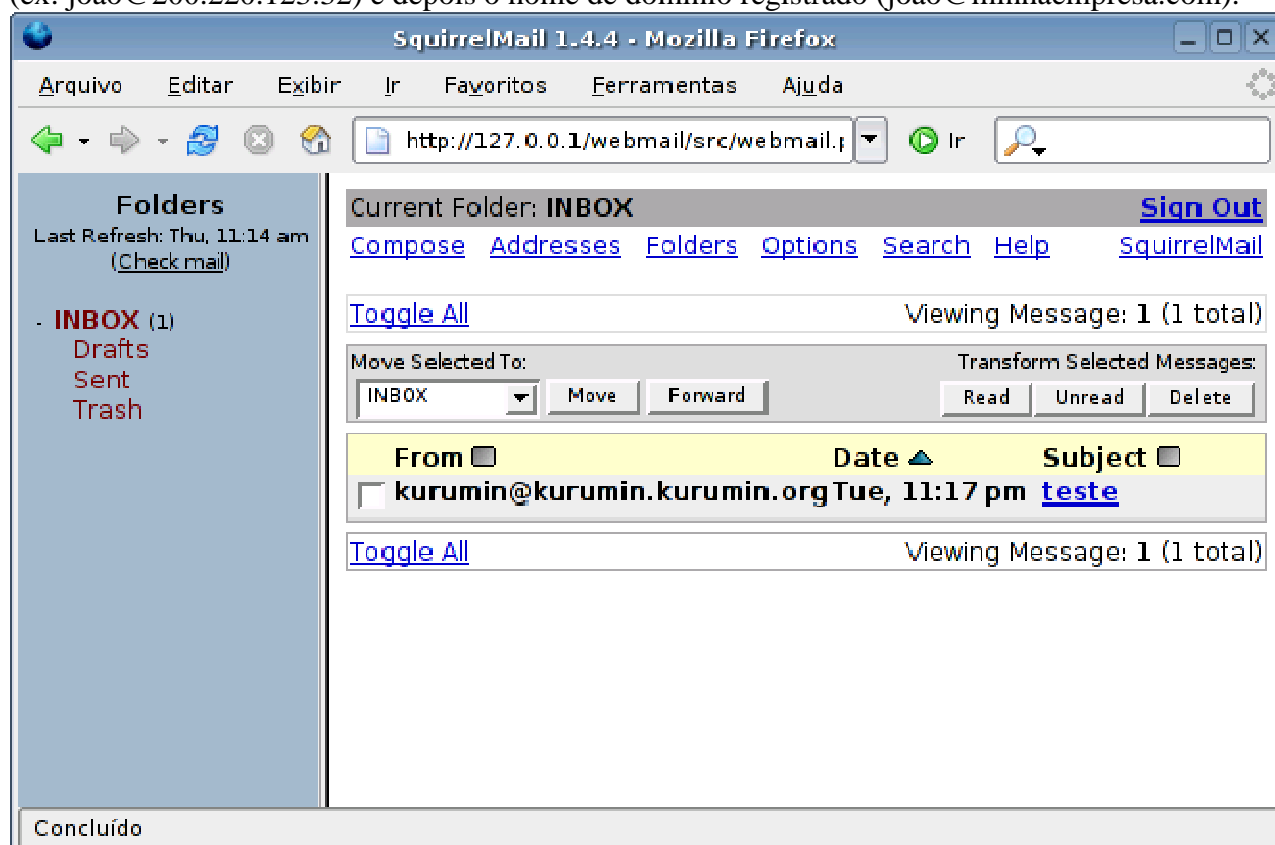
d) Acesse a opção **D** no menu e indique qual servidor IMAP está utilizando. Lembre-se de que no exemplo estamos usando o Courier. Definir corretamente o servidor usado aqui permite que o Squirrelmail ative uma série de otimizações para cada servidor específico, que melhoram consideravelmente o desempenho. Ao terminar, use a opção **S** para salvar e **Q** para sair.

O Squirrelmail não é um serviço, ele é apenas uma aplicação que roda dentro do Apache. Depois de instalar os arquivos, acesse o endereço "<http://127.0.0.1/webmail>" e você verá a tela de login do Squirrelmail.



Se algo der errado neste ponto, verifique a instalação do Apache, se o suporte a PHP está realmente funcionando (lembre-se de que além de instalar o pacote, é necessário incluir a linha "LoadModule php4_module /usr/lib/apache/1.3/libphp4.so" ou similar no arquivo de configuração do Apache) e se a pasta de instalação, o link ou a entrada no arquivo de configuração para vincular a pasta webmail/ com a pasta onde estão os arquivos estão corretos.

Depois de logado, faça alguns testes, verificando se consegue mandar e-mails para contas em outros servidores e se consegue mandar e-mails de um usuário local para outro. Se o servidor já estiver disponível na internet, experimente enviar um e-mail (a partir de outra conta) para ele, usando inicialmente o endereço IP (ex: joao@200.220.123.32) e depois o nome de domínio registrado (joao@minhaempresa.com).



Caso a tela de login funcione, mas você tenha problemas ao logar, verifique as permissões de acesso da pasta

de instalação do Squirrelmail, veja se ela não está com permissão de leitura apenas para o root. Lembre-se de que na maioria das distribuições o Apache roda sob o usuário "apache" ou "daemon" e não sob o usuário root, o que é inseguro.

Verifique também a configuração do Squirrelmail, veja se o serviço "**courier-imap**" está realmente inicializado. Observe que no Squirrelmail o login de acesso é apenas o nome do usuário (ex: joao) e não o endereço de e-mail completo. Ao configurar um servidor simples, onde as contas de acesso do sistema são usadas no servidor de e-mail, as senhas também são as mesmas.

Em algumas distribuições, depois de instalar o pacote courier-imap, é necessário rodar o comando "pw2userdb" para que as contas de usuários do sistema sejam corretamente incluídas como logins de acesso no servidor de e-mails. Verifique isso e reinicie o courier-imap novamente.

Uma última pegadinha é que, para que o servidor IMAP funcione, é necessário que exista um diretório chamado "Maildir" dentro do home de cada usuário, onde são armazenadas as mensagens. Este diretório contém uma estrutura própria e é criado usando o comando "maildirmake". Normalmente ele é criado automaticamente ao instalar os pacotes usados nas distribuições. Mas, em algumas, este procedimento precisa ser feito manualmente. É mais uma coisa que pode dar errado.

Se isso for necessário no seu caso, comece criando o diretório para o seu próprio usuário, ou o que for usar para testar o webmail:

```
$ maildirmake ~/Maildir
```

Execute agora o comando que cria a pasta dentro do diretório /etc/skel, de forma que os diretórios home de todos os novos usuários criados daqui em diante já sejam criados com ele:

```
# maildirmake /etc/skel/Maildir
```

Autenticando os clientes e Ativando o TLS

Originalmente, o Postfix determina os clientes que estão autorizados a enviar e-mails através do seu servidor de acordo com a configuração da linha "mynetworks", dentro do arquivo main.cf. Usando a linha "mynetworks = 127.0.0.0/8" ou "mynetworks = 127.0.0.1" o Postfix aceita apenas e-mails enviados a partir do próprio servidor, uma configuração ideal se os usuários enviam os e-mails através de um webmail instalado no próprio servidor, sem SMTP externo.

Você pode também permitir o envio a partir de qualquer micro da rede local, usando algo como "mynetworks = 192.168.0.0/24". O problema surge quando você precisa permitir o envio de e-mails para usuários espalhados pela web, conectados via ADSL, modem ou outras modalidades de conexão com IP dinâmico.

Imagine, por exemplo, o caso de um provedor de acesso que precisa permitir que seus usuários enviem e-mails usando seu SMTP, mesmo quando eles estiverem acessando através de outro provedor.

Você não pode simplesmente permitir o envio a partir de qualquer endereço, caso contrário seu servidor vai ser rapidamente descoberto pelos spammers, que começarão a utilizar toda a sua banda para enviar suas tentadoras ofertas de produtos. Pior, depois de algum tempo, seu servidor vai acabar caindo nas listas negras de endereços usados para envio de spam, fazendo com que seus próprios e-mails válidos passem a ser recusados por outros servidores.

A solução, nesse caso, é passar a autenticar os usuários, como faz a maioria dos provedores. Usamos então o SASL, que no Debian (Etch ou Sid) pode ser instalado via apt-get:

```
# apt-get install libsasl2 sasl2-bin libsasl2-modules libdb3-util procmail
```

Depois de instalar os pacotes, abra o arquivo `"/etc/default/saslauthd"`, onde vão as opções de inicialização do autenticador. O primeiro passo é substituir a linha `"START=no"` por:

```
START=yes
```

Adicione (ou modifique) também a linha:

```
MECHANISMS="pam"
```

Isso faz com que ele seja inicializado durante o boot e aceite a autenticação dos usuários.

Continuando, crie (ou edite) o arquivo `"/etc/postfix/sasl/smtpd.conf"` de forma que ele contenha apenas as linhas:

```
pwcheck_method: saslauthd  
mech_list: plain login
```

O pacote do Postfix usado no Debian Etch e no Ubuntu 6.10 (ou mais recente) e em outras distribuições derivadas deles, roda dentro de um chroot (ou jaula), o que melhora bastante a segurança, impedindo que qualquer eventual problema de segurança fique restrito ao servidor de e-mails, sem afetar o resto do sistema. Você notará que dentro da pasta `"/var/spool/postfix"` estão não apenas os diretórios com as filas de mensagens, mas também binários e bibliotecas de que o postfix precisa para funcionar.

O problema é que de dentro do seu chroot, o Postfix não tem acesso ao saslauthd, fazendo com que a autenticação dos usuários não funcione. O próprio saslauthd é necessário por que o Postfix (mesmo ao rodar fora do chroot) não tem acesso aos arquivos de senha do sistema e por isso não é capaz de autenticar os usuários por si só.

Para resolver este problema, precisamos criar a pasta `"/var/spool/postfix/var/run/saslauthd"`, utilizada pelo Postfix dentro do chroot e configurar o sasl para utilizá-la no lugar da pasta padrão. Desta forma, o Postfix consegue se comunicar com ele.

Este tipo de precaução de segurança parece algo complicado e desnecessário à primeira vista, mas é justamente por causa de truques como este que os servidores Linux acabam sendo tão seguros. Para começo de conversa, o Postfix é por si só bastante seguro. Mas, como os servidores de e-mail são um ponto comum de ataque, ele fica isolado dentro do chroot de forma que, mesmo na remota possibilidade de um cracker conseguir obter controle sobre o Postfix através um exploit remoto, ele não poderia fazer muita coisa. Para completar, o Postfix roda dentro de privilégios muito limitados, de forma que, mesmo que o cracker tenha muita sorte e a improvável falha de segurança no Postfix seja combinada com uma falha no sistema que o permita escapar do chroot, ele ainda assim não conseguiria fazer muita coisa. ;)

Comece criando o diretório:

```
# mkdir -p /var/spool/postfix/var/run/saslauthd
```

Abra agora o arquivo `"/etc/default/saslauthd"` (o mesmo onde substituímos o `"START=no"` por `"START=yes"`) e substitua a linha

```
OPTIONS="-c"
```

por:

```
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"
```

Reinicie o serviço para que as alterações entrem em vigor:

```
# /etc/init.d/saslauthd restart
```

Isso faz com que o SASL passe a utilizar o diretório dentro do chroot e o Postfix tenha acesso ao saslauthd e possa assim autenticar os usuários através dele. Note que o `/var/spool/postfix` é o diretório onde está o chroot. Esta é a localização padrão no Debian; ao usar outra distribuição, verifique se não está sendo usado outro diretório.

Só para garantir, adicione o postfix ao grupo sasl:

```
# adduser postfix sasl
```

Isso completa a configuração do SASL.

O passo seguinte é a configuração do Postfix. Abra o arquivo `/etc/postfix/main.cf` e adicione as linhas abaixo no final do arquivo. Ao reciclar um arquivo de configuração anterior, verifique se esta configuração já não foi incluída em outro ponto do arquivo:

```
smtpd_sasl_local_domain =  
smtpd_sasl_auth_enable = yes  
smtpd_sasl_security_options = noanonymous  
broken_sasl_auth_clients = yes  
smtpd_recipient_restrictions = permit_sasl_authenticated,  
permit_mynetworks,  
reject_unauth_destination  
smtpd_tls_auth_only = no
```

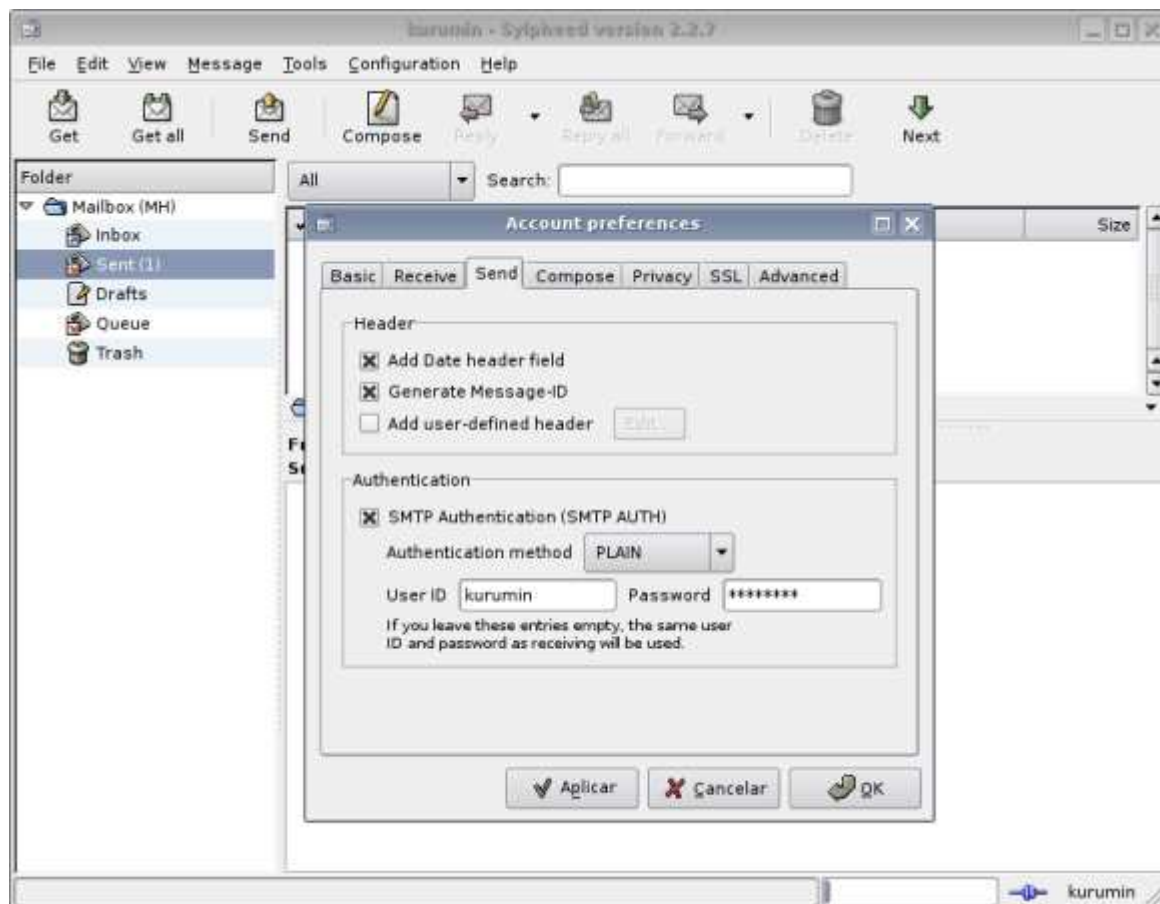
Feito isso, reinicie o Postfix para que as alterações entrem em vigor:

```
# /etc/init.d/postfix restart
```

Por enquanto, o servidor suporta apenas autenticação em texto puro, sem encriptação. Este é o sistema "clássico", ainda usado por muito provedores de acesso, mas que possui problemas óbvios de segurança, já que alguém que consiga sniffar a rede local, poderia capturar as senhas dos usuários no momento em que eles tentassem baixar os e-mails.

Para testar, configure um cliente de e-mails qualquer para utilizar o endereço IP do servidor como SMTP (aqui estou usando o Sylspeed) e, nas configurações, ative a opção de autenticação para o servidor SMTP e escolha a opção "PLAIN" (login em texto puro). Envie um e-mail de teste para confirmar se tudo está

funcionando.



Ativando o TLS

O TLS (Transport Layer Security) adiciona segurança ao nosso sistema de autenticação, permitindo que os usuários possam baixar os e-mails sem medo, mesmo ao acessar a partir de redes públicas.

Em algumas distribuições (como no Debian Sarge), você precisa instalar o pacote "postfix-tls". Nas demais (incluindo o Debian Etch, que é a versão atual), ele já vem integrado ao pacote principal do Postfix.

O TLS trabalha utilizando um conjunto de chaves de encriptação e certificados, usados para criar o túnel encriptado e garantir a segurança da seção. O primeiro passo é criar este conjunto de arquivos.

Acesse o diretório "/etc/postfix/ssl" (crie-o se não existir) e rode os comandos abaixo, um de cada vez e nesta ordem. Durante a geração das chaves, será solicitado que você informe uma passphrase, uma senha que pode conter entre 4 e 8191 caracteres. Administradores paranóicos costumam usar passphrases bem grandes, mas não exagere, pois você precisará confirmá-la algumas vezes durante o processo. Os comandos são:

```
# mkdir /etc/postfix/ssl
# cd /etc/postfix/ssl/
# openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
# chmod 600 smtpd.key
# openssl req -new -key smtpd.key -out smtpd.csr
# openssl x509 -req -days 730 -in smtpd.csr -signkey smtpd.key -out smtpd.crt
# openssl rsa -in smtpd.key -out smtpd.key.unencrypted
# mv -f smtpd.key.unencrypted smtpd.key
# openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days 730
```

O "730" usado nas linhas determina a validade dos certificados, em dias. No caso, estou criando certificados

válidos por dois anos. Depois deste prazo, os clientes começarão a receber um aviso ao se autenticarem, avisando que o certificado expirou e precisarei repetir o processo para atualizá-los. Se preferir, você pode usar um número mais alto, para gerar certificados válidos por mais tempo. Para gerar certificados válidos por 10 anos, por exemplo, substitua o "730" por "3650".

Continuando, abra novamente o arquivo **"/etc/postfix/main.cf"** e adicione as linhas abaixo (sem mexer nas linhas referentes ao SASL que adicionamos anteriormente):

```
smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
tls_random_source = dev:/dev/urandom
```

Reinicie o Postfix para que as alterações entrem em vigor:

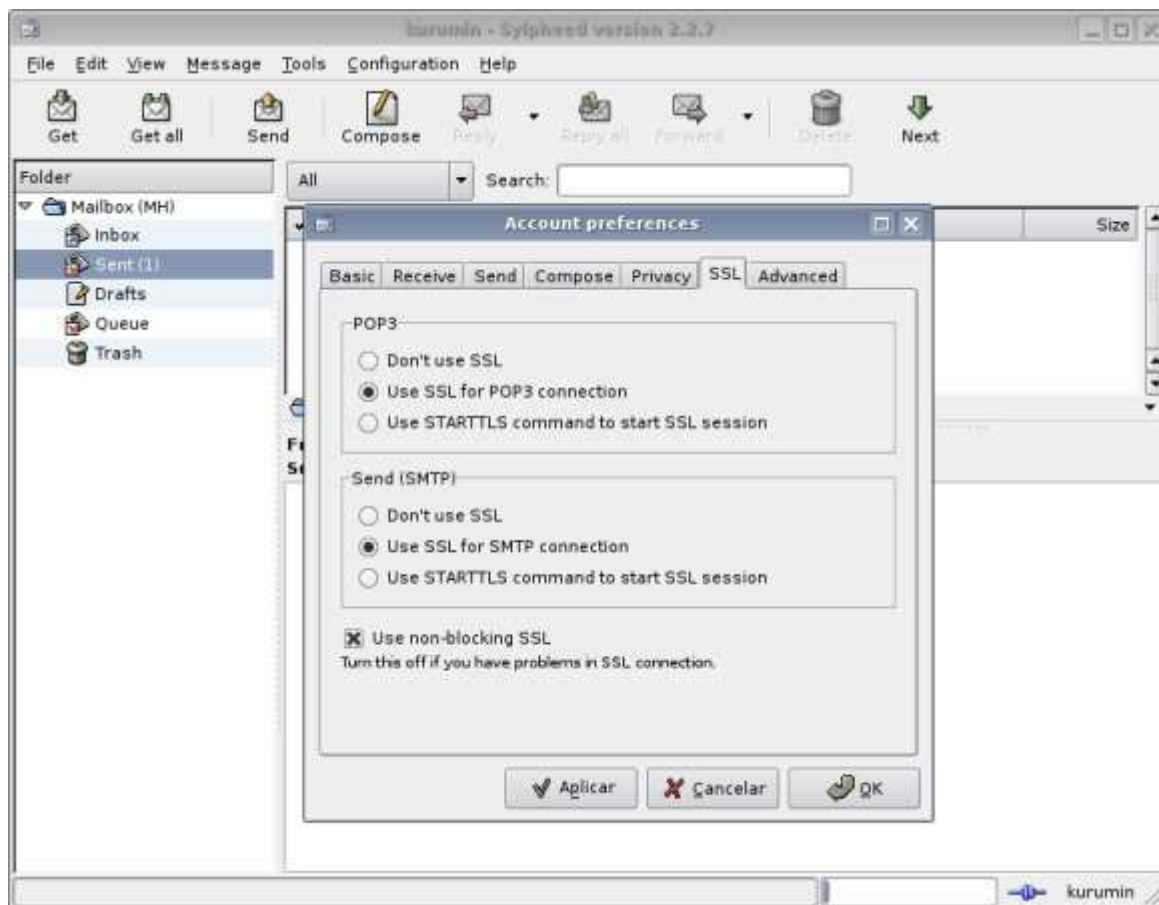
```
# /etc/init.d/postfix restart
```

Para que os clientes consigam se autenticar no servidor, é necessário instalar o pacote "courier-authdaemon" e o "courier-ssl", além dos pacotes courier-pop, courier-pop-ssl, courier-imap, courier-imap-ssl que vimos anteriormente. Você pode usar o comando abaixo para instalar de uma vez todos os pacotes necessários:

```
# apt-get install courier-authdaemon courier-base courier-imap courier-imap-ssl \
courier-pop courier-pop-ssl courier-ssl gamin libgamin0 libglib2.0-0
```

Para testar, ative o uso do SSL para o servidor SMTP dentro das preferências do seu cliente de e-mail. No caso do Thunderbird, por exemplo, marque a opção "Usar Conexão Segura > TLS" dentro do menu "Enviar", nas configurações da conta. O cliente de e-mail exibirá alguns avisos sobre a validade do certificado, o que é normal, já que estamos utilizando um certificado "self-signed", ou seja, um certificado "caseiro", que não é reconhecido por nenhuma autoridade certificadora. Empresas como a Verisign vendem certificados reconhecidos, mas os preços são proibitivos fora de grandes instalações.

Com o TLS, A autenticação continua funcionando da mesma forma, mas agora todos os dados são transmitidos de forma segura. Lembre-se de que ao instalar o courier, já ativamos também o suporte a SSL para o IMAP e POP3, de forma que você pode ativar ambas as opções no cliente de e-mail:



Aqui está um exemplo de arquivo **/etc/postfix/main.cf** completo, incluindo a configuração do SASL e do TLS:

```
# /etc/postfix/main.cf
```

```
myhostname = etch.kurumin.com.br
mydomain = kurumin.com.br
append_dot_mydomain = no
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = etch.kurumin.com.br, kurumin.com.br, localhost
relayhost =
mynetworks = 127.0.0.0/8
home_mailbox = Maildir/
mailbox_command =
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
message_size_limit = 20000000
mailbox_size_limit = 0

smtpd_sasl_local_domain =
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_sasl_authenticated,
permit_mynetworks,
reject_unauth_destination
```

```
smtpd_tls_auth_only = no
```

```
smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
tls_random_source = dev:/dev/urandom
```

O arquivo **/etc/default/saslauthd** (depois de removidos os comentários), ficaria:

```
START=yes
MECHANISMS="pam"
MECH_OPTIONS=""
THREADS=5
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"
```

O arquivo **/etc/postfix/sasl/smtpd.conf**, que vimos no início, continua com apenas as duas linhas:

```
pwcheck_method: saslauthd
mech_list: plain login
```

Adicionando um antivírus

Depois que o servidor de e-mails estiver funcionando, é interessante instalar um antivírus para proteger as estações Windows. Este é um detalhe interessante: existem vários bons antivírus para Linux, mas todos são destinados a justamente encontrar vírus for Windows em compartilhamentos de rede, páginas web e arquivos, e-mails, etc. Até hoje, o Linux (tanto como servidor, quanto desktop) tem se mantido como uma plataforma livre de vírus dignos de nota, daí a ausência de soluções neste sentido. A demanda simplesmente não existe.

Uma das melhores opções é o **Clamav**, que possui uma lista de definições atualizada com uma frequência muito grande e oferece um recurso de atualização automática. O Clamav escaneia as mensagens que passam pelo servidor, removendo as mensagens com arquivos infectados. Ele serve tanto para proteger clientes Windows da rede, quanto para reduzir o tráfego de mensagens inúteis.

Para instalar o antivírus, basta instalar o pacote "**clamav**". Ele não costuma mudar de nome entre as distribuições. No Debian, você precisa instalar também o pacote "**clamav-daemon**", em outras distribuições este componente faz parte do pacote principal.

Para utilizar o Clamav em conjunto com o Postfix, de forma que todos os e-mails passem primeiro pelo antivírus, e só depois sejam encaminhados para as caixas postais dos usuários, é necessário instalar também o pacote "**amavisd-new**".

O Amavisd "intercepta" as novas mensagens, entregando-as ao executável do Clamav. De acordo com a configuração, as mensagens com arquivos infectados podem ser simplesmente deletadas, ou colocadas em uma pasta de quarentena. Lembre-se de que quase todas as mensagens com arquivos infectados são enviadas automaticamente pelos vírus da moda, como uma forma de se espalharem, por isso não existe muito sentido em preservá-las. O Amavisd é um software complicado de instalar manualmente, é necessário alterar vários

scripts e arquivos de inicialização e configurar corretamente as permissões de várias pastas. Além de trabalhoso, o processo é muito sujeito a erros, por isso é sempre recomendável utilizar os pacotes incluídos nas distribuições, onde o trabalho já está feito.

A comunicação entre o Amavisd e o Clamav é feita automaticamente, mas é necessário configurar o Postfix para direcionar os novos e-mails para o Amavisd, para que o trio comece a trabalhar em conjunto. Para isso, é necessário adicionar as linhas abaixo no final do arquivo `"/etc/postfix/master.cf"` (note que este arquivo é diferente do `main.cf` que configuramos anteriormente). Estas linhas estão incluídas no arquivo `"/usr/share/doc/amavisd-new/README.postfix"`; você pode copiá-las a partir do arquivo, ao invés de escrever tudo:

```
smtp-amavis unix - - n - 2 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes

127.0.0.1:10025 inet n - n - - smtpd
-o content_filter=
-o local_recipient_maps=
-o relay_recipient_maps=
-o smtpd_restriction_classes=
-o smtpd_client_restrictions=
-o smtpd_helo_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks=127.0.0.0/8
-o strict_rfc821_envelopes=yes
-o smtpd_error_sleep_time=0
-o smtpd_soft_error_limit=1001
-o smtpd_hard_error_limit=1000
```

Adicione também a linha abaixo ao `"/etc/postfix/main.cf"`:

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

É preciso verificar também os arquivos `"/etc/clamav/clamd.conf"` e `"/etc/amavis/amavisd.conf"`. Em muitas distribuições eles são configurados corretamente ao instalar os pacotes, mas em outras é preciso fazer as alterações manualmente.

No arquivo `"/etc/clamav/clamd.conf"`, verifique se a linha abaixo está presente e descomentada:

```
LocalSocket /var/run/clamav/clamd.ctl
```

No arquivo `"/etc/amavis/amavisd.conf"`, verifique se as linhas abaixo estão descomentadas. Este arquivo inclui vários exemplos que permitem usar diferentes antivírus, por isso é um pouco extenso. Use a função de pesquisar do editor de textos para ir direto ao ponto:

```
### http://www.clamav.net/
['Clam Antivirus-clamd',
 \&ask_daemon, ["CONTSCAN {} \n", "/var/run/clamav/clamd.ctl"],
 qr/\bOK$/, qr/\bFOUND$/,
 qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

O último passo é fazer com que o Amavis tenha acesso aos arquivos de log e diretórios de trabalho do Clamav. No Debian e em muitas outras distribuições, basta adicionar o usuário **"clamav"** no grupo **"amavis"**, a solução mais rápida e limpa:

```
# adduser clamav amavis
```

Caso você esteja usando outra distribuição, onde essa primeira alteração não funcione, resta fazer do jeito sujo, alterando manualmente as permissões de acesso dos diretórios:

```
# chown -R amavis:amavis /var/clamav
# chown -R amavis:amavis /var/log/clamav
# chown -R amavis:clamav /var/run/clamav
```

Depois de terminar a configuração, reinicie todos os serviços, de forma que tudo entre em vigor:

```
# /etc/init.d/postfix restart
# /etc/init.d/clamav-daemon restart
("/etc/init.d/clamd restart" no Fedora e Mandriva)
# /etc/init.d/clamav-freshclam restart
("/etc/init.d/freshclam restart" no Fedora e Mandriva)
# /etc/init.d/amavis restart
```

Experimente tentar enviar agora um e-mail contendo um arquivo infectado qualquer para uma conta qualquer do seu servidor. O e-mail simplesmente não vai chegar ao destino. O arquivo `/usr/share/doc/amavisd-new/README.postfix` contém uma string de texto que dispara o antivírus e pode ser usada para simular um e-mail infectado.

Checando o conteúdo do arquivo `/var/log/clamav/clamav.log`, você verá uma entrada indicando que um e-mail infectado foi encontrado:

```
Thu Ago 26 19:55:49 2005 -> /var/lib/amavis/amavis-20050526T195549-07338/parts/part-00001: Eicar-Test-Signature FOUND
```

O próximo passo é instalar o Spamassassin que funciona como um filtro antispam automático, que utiliza uma blacklist com endereços IP e conteúdos de mensagem catalogados. Esta lista funciona de forma semelhante à de um programa antivírus, é atualizada pela equipe de desenvolvimento e atualizada de forma automática. Para instalar:

```
# apt-get install spamassassin
```

No Debian, por padrão, ele fica inativo depois de instalado, talvez como uma precaução para evitar consumir muitos recursos em micros onde ele é instalado acidentalmente junto com outros servidores. Para ativá-lo, edite o arquivo `/etc/default/spamassassin`, mudando a opção `ENABLED=0` para `ENABLED=1`:

```
# Change to one to enable spamd
ENABLED=1
```

Para finalizar, inicie o serviço **"spamassassin"** (ou **"spamd"** em muitas distribuições):

```
# /etc/init.d/spamassassin start
```

Falta agora configurar o Amavis para utilizar também o Spamassassin ao receber novas mensagens. Agora os e-mails passarão pelos dois filtros, seqüencialmente.

Abra novamente o arquivo: `/etc/amavis/amavisd.conf`. Na seção 1, por volta da linha 160, **comente** (#) a linha:

```
@bypass_spam_checks_acl = qw( . );
```

Essa linha desativa a checagem de spam. Ela vem descomentada por padrão, pois nem todo mundo utiliza o Amavis em conjunto com o Spamassassin. Ao comentá-la, a checagem é ativada.

Na seção 4, por volta da linha 400, procure pela linha: `$final_spam_destiny =`.

Essa linha configura o que será feito com as mensagens marcadas como spam. Ela tem três valores possíveis: **D_PASS** – Entrega a mensagem normalmente, incluindo apenas a palavra "SPAM" no subject. Isso permite que os próprios usuários configurem o filtro local do leitor de e-mails para remover as mensagens caso estejam incomodando. Nenhum filtro antispam é perfeito, sempre algumas mensagens legítimas acabam sendo marcadas como spam. Esta opção minimiza o problema, deixando a remoção das mensagens por conta dos usuários.

D_DISCARD – Descarta a mensagem. Esta é a opção mais usada, mas ao mesmo tempo a mais perigosa, pois mensagens "boas" marcadas como spam vão simplesmente sumir, sem deixar nenhum aviso ao remetente ou destinatário.

D_REJECT – Esta terceira opção também descarta a mensagem, mas envia uma notificação ao emissor. Isso permite que o emissor de uma mensagem "boa", acidentalmente classificada como spam, receba um aviso de que ela foi descartada e tenha a oportunidade de reenviá-la novamente de outra forma. Mas, por outro lado, como a maioria dos spams são enviados a partir de endereços falsos, isso acaba sendo um desperdício de banda.

Ao usar a opção que descarta as mensagens, a opção fica:

```
$final_spam_destiny = D_DISCARD
```

Mais adiante, na seção 7, por volta da linha 1100, você encontrará mais opções relacionadas ao Spamassassin.

Finalmente, é preciso transferir o ownership dos arquivos do Spamassassin para o Amavis (assim como no caso do Clamav), para que ele possa executar corretamente suas funções:

```
# chown -R amavis:amavis /usr/share/spamassassin
```

Não se esqueça de reiniciar os serviços com o comando:

```
# /etc/init.d/spamassassin restart  
("/etc/init.d/spamd restart" em muitas distribuições)  
# /etc/init.d/amavis restart
```

Além de trabalhar em conjunto com o Postfix (ou com o Qmail ou mesmo Sendmail), o Spamassassin pode ser usado diretamente pelo Evolution ou Kmail para filtrar os e-mails localmente. Você pode ver mais detalhes de como configurar o cliente no: <http://brlinux.linuxsecurity.com.br/tutoriais/000665.html#000665>.

Configurando o DNS Reverso

O DNS reverso é um recurso que permite que outros servidores verifiquem a autenticidade do seu servidor, checando se o endereço IP atual bate com o endereço IP informado pelo servidor DNS. Isso evita que alguém utilize um domínio que não lhe pertence para enviar spam, por exemplo.

Qualquer um pode enviar e-mails colocando no campo do remetente o servidor do seu domínio, mas um servidor configurado para checar o DNS reverso vai descobrir a farsa e classificar os e-mails forjados como spam.

O problema é que os mesmos servidores vão recusar seus e-mails, ou classificá-los como spam, caso você não configure seu servidor DNS corretamente para responder às checagens de DNS reverso. Chegamos a um ponto em que o problema do spam é tão severo que quase todos os servidores importantes fazem esta checagem, fazendo com que, sem a configuração, literalmente metade dos seus e-mails não seja entregue.

O primeiro passo é checar os arquivos `/etc/hostname` e `/etc/hosts` (no servidor), que devem conter o nome da máquina e o domínio registrado.

O arquivo `"/etc/hostname"` deve conter apenas o nome da máquina, como em:

```
servidor
```

No Fedora e em algumas outras distribuições, o nome da máquina vai dentro do arquivo `"/etc/sysconfig/network"`.

O arquivo `"/etc/hosts"` deve conter duas entradas, uma para a interface de loopback, o 127.0.0.1 e outra para o IP de internet do seu servidor, que está vinculado ao domínio, como em:

```
127.0.0.1 localhost.localdomain localhost
64.234.23.12 etch.kurumin.com.br etch
```

A partir daí, falta adicionar a zona reversa no bind complementando a configuração do domínio, que já fizemos. Começamos adicionando a entrada no `"/etc/bind/named.conf"` ou `"/etc/bind/named.conf.local"`:

```
zone "23.234.64.in-addr.arpa" {
type master;
file "/etc/bind/db.kurumin.rev";
};
```

No nosso exemplo, o endereço IP do servidor é 64.234.23.12. Se retiramos o último octeto e escrevemos o restante do endereço de trás pra frente, temos justamente o "23.234.64" que usamos no registro reverso. A terceira linha indica o arquivo em que a configuração do domínio reverso será salva. Nesse caso, indiquei o arquivo `"db.kurumin.rev"`, mas você pode usar qualquer nome de arquivo.

Este arquivo `"db.kurumin.rev"` é bem similar ao arquivo com a configuração do domínio. As três linhas iniciais são idênticas (incluindo o número de sincronismo), mas ao invés de usar o "A" para relacionar o domínio e cada subdomínio ao IP correspondente, usamos a diretiva "PTR" para relacionar o endereço IP de cada servidor ao domínio (é justamente por isso que chamamos de DNS reverso ;).

No primeiro arquivo, usamos apenas os três primeiros octetos do endereço (a parte referente à rede), removendo o octeto final (o endereço do servidor dentro da rede). Agora, usamos apenas o número omitido da primeira vez.

O IP do servidor é "64.234.23.12"; removendo os três primeiros octetos, ficamos apenas com o "12". Temos também o endereço do DNS secundário, que é 64.234.23.13, de onde usamos apenas o "13". Relacionando os dois a seus respectivos domínios, o arquivo fica:

```
@ IN SOA etch.kurumin.com.br. hostmaster.kurumin.com.br. (
2006040645 3H 15M 1W 1D )
NS etch.kurumin.com.br.
NS ns1.kurumin.com.br.
12 PTR kurumin.com.br.
13 PTR ns1.kurumin.com.br.
```

Caso você não esteja usando um DNS secundário, é só omitir as linhas referentes a ele, como em:

```
@ IN SOA etch.kurumin.com.br. hostmaster.kurumin.com.br. (
2006040645 3H 15M 1W 1D )
NS etch.kurumin.com.br.
12 PTR kurumin.com.br.
```

Depois de terminar, reinicie o Bind e verifique usando o **dig**. Comece checando o domínio, como em:

```
# dig kurumin.com.br
```

Na resposta, procure pela seção "ANSWER SECTION", que deverá conter o IP do servidor, como configurado no bind:

```
:: ANSWER SECTION:  
kurumin.com.br. 86400 IN A 64.234.23.12
```

Faça agora uma busca reversa pelo endereço IP, adicionando o parâmetro "-x", como em:

```
# dig -x 64.234.23.12
```

Na resposta você verá:

```
:: ANSWER SECTION:  
12.23.234.64.in-addr.arpa. 86400 IN PTR kurumin.com.br.
```

Ou seja, com o DNS reverso funcionando, o domínio aponta para o IP do servidor e o IP aponta para o domínio, permitindo que os outros servidores verifiquem a autenticidade do seu na hora de receber e-mails provenientes do seu domínio.

Lembre-se que seus e-mails podem ser classificados como spam também se seu IP estiver marcado em alguma blacklist. Você pode verificar isso rapidamente no <http://rbls.org/>.

Você vai notar, por exemplo, que praticamente qualquer endereço IP de uma conexão via ADSL ou modem vai estar listado, muitas vezes "preventivamente", já que é muito comum que conexões domésticas sejam usadas para enviar spam. É recomendável verificar periodicamente os IP's usados pelo seu servidor, além de verificar qualquer novo IP ou link antes de contratar o serviço.

Mais uma consideração importante sobre a configuração do DNS reverso é que você precisa ter autoridade sobre a faixa de IP's para que a configuração funcione.

Quando você aluga um servidor dedicado, é de praxe que receba uma pequena faixa de endereços IP, configurada usando uma máscara complexa, como a "**255.255.255.248**", onde você fica com uma faixa de 5 IP's diferentes (na verdade são 8, mas destes apenas 6 são válidos e um é usado como default gateway, para que seu servidor possa acessar a internet através da rede do datacenter) como, por exemplo, do "72.213.43.106" até o "72.213.43.110".

Isso também é comum ao alugar um link dedicado, onde (dependendo do plano) você recebe uma faixa completa, com 254 IP's, ou uma faixa reduzida, com máscara "255.255.255.240" (14 IP's), "255.255.255.248" (6 IP's) ou "255.255.255.252" (2 IP's).

Em qualquer um dos casos, o fornecedor deve delegar a autoridade sobre a sua faixa de endereços, para que a configuração que vimos aqui funcione. Sem isto, é o servidor deles quem responde pela sua faixa, retornando um erro qualquer, sem que seu servidor DNS tenha chance de fazer seu trabalho.

Se você alugou um servidor ou um link dedicado e percebeu que a autoridade sobre a faixa não foi delegada, minha primeira sugestão é que você troque de fornecedor, pois essa é uma configuração básica. Se não fazem nem a delegação corretamente, é provável que o serviço deixe a desejar em outros aspectos. Se isso não for possível, entre em contato com eles cobrando a configuração.

Alguns provedores preferem configurar eles mesmos o DNS reverso para seu domínio. Nesse caso, vão

apenas lhe pedir a configuração e ativar o reverso no DNS deles. Essa solução não é ideal, pois você vai precisar entrar em contato com o suporte cada vez que precisar fazer uma alteração, mas é melhor do que nada. Essa é também a única opção em planos em que você recebe um único IP fixo, ao invés de uma faixa de endereços.

No caso de planos de acesso doméstico, onde você recebe um único IP, quase sempre é impossível configurar o DNS reverso, pois você não tem autoridade sobre a faixa de IP's e a operadora não vai querer fazer a configuração para você sem que você pague mais um plano empresarial.