Tutorial POSTFIX

Centro de Informática de Ribeirão Preto - CIRP

MSc. Eng. Ali Faiez Taha

Sumário

Capítulo 1

Servidor de E-Mails - Postfix

1.1 Introdução

O correio eletrônico é um dos serviços mais utilizados na Internet, e cada vez mais pessoas e empresas utilizam-no para trocar informações de maneira rápida e eficiente.

Neste tutorial será visto como funciona e como implementar um serviço de correio eletrônico. Será visto também como implementar um **webmail** (um serviço que permite que os usuários acessem as suas mensagens através de um navegador Internet), como criar filtros para barrar mensagens não solicitadas, etc.

1.2 Funcionamento do Correio Eletrônico

Antes de implementar um serviço de correio eletrônico é importante que o administrador entenda como funciona a troca de mensagens, seja na Internet, seja em uma rede local. Para uma simples troca de mensagens entre dois usuários, pode ser necessária a utilização de vários protocolos e de várias aplicações.

Um usuário que queira enviar uma mensagem para outro utilizará um aplicativo cliente de e-mail, também conhecido como **MUA**, ou **Agente de Mensagens do Usuário**. Ao terminar de redigir a sua mensagem o MUA enviará a mensagem a um **MTA (Agente Transportador de Mensagens)** que se encarregará então de entregar a mensagem ao **MTA** do destinatário, caso ele se encontre em outra máquina ou simplesmente colocar a mensagem na caixa postal do destinatário, caso ele se encontre no mesmo servidor.

A transferência da mensagem entre o **MUA** e o **MTA** se efetua utilizando-se um protocolo chamado **SMTP,** ou **Protocolo Simples de Transferência de Mensagens**. O protocolo **SMTP** será utilizado também entre o **MTA** do remetente e o **MTA** do destinatário.

O servidor de **e-mail** do destinatário, ao receber uma mensagem para um dos seus usuários, simplesmente a coloca na caixa postal deste usuário.

Se o usuário possui uma conta **shell** neste servidor ele poderá ler os seus **e-mails** direto no servidor, caso contrário o usuário deverá transferir suas mensagens para sua máquina a fim de lê-las com o seu cliente de **e-mail**. A transferência de mensagens recebidas entre o servidor e o cliente de **e-mail** requer a utilização de outros programas e protocolos. Usualmente é utilizado para este fim o protocolo **POP**, Protocolo de "Agência" de Correio, que recebe este nome por agir como uma agência de correios mesmo, que guarda as mensagens dos usuários em caixas postais e aguarda que estes venham buscar suas mensagens. Outro protocolo que pode ser utilizado para este mesmo fim é o **IMAP**, **Protocolo para Acesso de Mensagens via Internet**, que implementa além das funcionalidades fornecidas pelo **POP** muitos outros recursos. Os protocolos **POP** e **IMAP** são protocolos para recebimentos de mensagens, ao contrário do protocolo **SMTP** que serve para enviar mensagens, logo, possuem funcionalidades diferenciadas, como, por exemplo, autenticação do usuário.

Para a utilização dos protocolos **POP** e **IMAP** é necessário a instalação do servidor apropriado, que vai ser o responsável por atender as solicitações do cliente de **e-mail** por novas mensagens. O recebimento de mensagens pelo cliente se dá através da solicitação do **MUA** do usuário ao seu servidor de **e-mail**, que após a autenticação do usuário vai informar se existem mensagens em sua caixa postal e quantas são. A seguir o **MUA** solicita a transferência das mensagens para a máquina local, finalizando assim o processo de troca de mensagens entre dois usuários.

1.3 Postfix

O **Postfix** é o **MTA** padrão de muitas distribuições Unix/Linux. (www.postfix.org)

O **Postfix** vem se consolidando como uma alternativa ao **Sendmail** (www.sendmail.org) em razão de suas características tais como:

maior robustez, melhor desempenho e maior facilidade na manutenção e configuração.

Além do mais o **Postfix** é capaz de emular várias funções do **Sendmail** evitando assim modificações nas aplicações que utilizam o **Sendmail**.

Outra característica importante do **Postfix** é a sua construção modular facilitando a manutenção do código e permitindo a implementação de novas funcionalidades mais facilmente.

Para uma implementação bem-sucedida do Postfix é necessário:

uma interface de rede instalada e configurada;

um servidor DNS instalado e configurado. A instalação no Debian Linux é bastante simples e é feita através do comando:

$> apt-get install postfix
$> apt-get install mailx —> quem realmente envia os E-Mails

Opcionais :
$> apt-get install procmail —> Filtro de E-Mail
$> apt-get install squirrelmail —> WEBMAIL
$> apt-get install php5 —> Linguagem PHP

Outros WEBMAILs :
Openwebmail - http://www.openwebmail.org
IMP Horde - http://www.horde.org/imp

1.4 Configuração do Postfix

Os arquivos de configuração do **Postfix** estão localizados no diretório **/etc/postfix.** No FreeBSD o local padrão é **/usr/local/etc/postfix**

Os principais arquivos são : **main.cf e master.cf**

As configurações devem ser modificadas de acordo com os seguintes itens:
hostname, dominio, Servidor DNS, arquivos de controle.

Os detalhes das configurações destes arquivos estão no URL:
http://www.postfix.org/BASIC_CONFIGURATION_README.html

1.4.1 Configurações básicas

Configurações básicas no arquivo main.cf **Domínio de rede** a ser utilizado para saída de E-Mails: **myhostname**

Domínio de rede para receber E-Mails: **mydestination**

Clientes que podem fazer **Relay: mynetworks**
Quais são os **Relays** de destino: **relay_domains**
Método de envio: Direto ou Indireto : **relayhost**
Problemas que devem ser enviados ao **Postmaster: notify_classes**
Configurações em condições de rede com **Proxy ou NAT: proxy_interfaces**
Registros de **logs** das atividades do servidor: **/etc/syslog.conf, itens mail.err e mail.debug**
Necessidades para o servidor trabalhar em modo **chroot: no arquivo master.cf**
Nomes de Hosts, de Domínios e identificação de Rede : **myhostname, mydomain e inet_interfaces**

Configurações mais detalhadas: Informação de caminho local
Área de armazenamento temporária (**spool**), e o caminho para os binários dos programas e serviços que compõem o Postfix.

Proprietário da fila e processo
O **Postfix** roda como um usuário comum do sistema e com permissões de usuários comuns, tornando-o assim mais seguro. São configurados o usuário **dono** dos arquivos de armazenamento temporário de mensagens e as permissões do programa.

Nome da domínio e da máquina
Nome da máquina (**nome + domínio**) e o **domínio**. Normalmente esses valores são obtidos automaticamente, no entanto pode ser necessário configurá-los manualmente, principalmente se a máquina em questão possui um nome e um apelido. Este nome e o domínio serão utilizados como padrão para o tráfego de mensagens.

Enviando mail
O nome padrão a ser utilizado ao serem enviadas mensagens a partir dessa máquina, ou seja, especifica a máquina ou domínio de onde **e-mails** postados localmente parecerão ter vindo. A configuração padrão é utilizar o nome da máquina, e assim, um usuário, ao enviar uma mensagem, seria identificado como **usuario@maquina**, por exemplo. É possível utilizar apenas o domínio (**usuario@dominio**), ou ainda, um outro nome qualquer. Note que isso só afeta o nome padrão adotado por programas clientes de **e-mail** desta máquina, e que os usuários podem configurar os seus programas de e-mail para enviar mensagens utilizando outro nome de domínio.

Recebendo mail
Regras básicas para recebimento de mensagens. Configurar os endereços de interface de rede pelos quais o sistema receberá mensagens. Por padrão o Postfix aceita mensagens por todos os endereços. No campo Destino (mydestination) insira os endereços ou informe um arquivo contendo estes endereços para os quais essa máquina é o destino final. Por exemplo, se o seu domínio é **minhaorganizacao** e esta é a máquina encarregada de receber as mensagens deste domínio, insira "minhaorganizacao" neste campo, mas atenção, não insira aqui os **domínios virtuais** hospedados nessa máquina pois essa configuração é feita em outro local. A configuração inadequada deste campo causará a mensagem de erro **"Mail for xx.xx.xx loops back to myself"** (**mensagem para xx.xx.xx retorna para mim**), pois o **Postfix** não está configurado para ser o destino final desse endereço.

Rejeitando usuários locais desconhecidos
O administrador poderá indicar um arquivo que contém os usuários que pertencem ao domínio e que podem receber mensagens. Não é preciso preencher nada nessa opção para uma configuração básica.

Controle de depuração
Configurações relativas às mensagens de registro utilizadas pelo **Postfix** para a resolução de problemas. Para uma configuração simples utilize os valores padrões.

1.4.2 Exemplos de configurações

- **Postfix como um servidor Standalone:**

/etc/postfix/main.cf:

```
# Optional: send mail as user@domainname instead of user@hostname.
#myorigin = $mydomain
# Optional: specify NAT/proxy external address.
#proxy_interfaces = 1.2.3.4
# Don't relay mail from other hosts.
mynetworks_style = host
relay_domains =
```

- **Postfix como null client:**

É um servidor que somente envia E-Mails. Não recebe E-Mails pela rede e não envia E-Mail localmente. Usar POP, IMAP e NFS para acessar os Mailbox.

```
1 /etc/postfix/main.cf:
2 myorigin = $mydomain
3 relayhost = $mydomain
4 inet_interfaces = 127.0.0.1
5 local_transport = error:local delivery is disabled
6
7 /etc/postfix/master.cf:
8 Comment out the local delivery agent entry
```

Traduzindo:

Linha 2: Envia E-Mail como "usuario@exemplo.com", ao invés de "usuario@nullclient.exemplo.com"

Linha 3: Redireciona todos E-Mails para o servidor de E-Mails responsável pelo domínio "exemplo.com"

Linha 4: Não aceita E-Mails da rede

Linhas 5-8: Desabilita o envio de E-Mails localmente. Todos os E-Mails vão pelo servidor especificado na linha 3

- **Postfix numa rede local**

Tráfego de E-Mails apenas na rede local (10.0.0.0/24)

```
1 /etc/postfix/main.cf:
2 myorigin = $mydomain
3 mynetworks = 127.0.0.0/8 10.0.0.0/24
4 relay_domains =
5 # Optional: forward all non-local mail to mailhost
6 #relayhost = $mydomain
```

Traduzindo:

Linha 2: Envia E-Mails como "usuario@exemplo.com"

Linha 3: Especifica as redes locais confiáveis

Linha 4: Não faz relay de outras redes não confiáveis

Linha 6: Necessário se não tem acesso à Internet

- **Exemplo de configuração para MAILHOST**

A máquina envia E-Mails como "usuario@exemplo.com" e o destino final como "usuario@hostname.exemplo.com" assim como "usuario@exemplo.com"

```
1 DNS:
2 example.com IN MX 10 mailhost.example.com.
```

3
4 /etc/postfix/main.cf:
5 myorigin = $mydomain
6 mydestination = $myhostname localhost.$mydomain localhost $mydomain
7 mynetworks = 127.0.0.0/8 10.0.0.0/24
8 relay_domains =
9 # Optional: forward all non-local mail to firewall
10 #relayhost = [firewall.example.com]

Traduzindo:

Linha 2: Configuração de DNS. Envia E-Mails para o domínio "exemplo.com" significa enviar para a máquina mailhost.exemplo.com. Lembre-se de especifica o ponto (".") no final da linha.

Linha 5: Envia E-Mails como "usuario@exemplo.com"

Linha 6: Este host é o destino final dos E-Mails para o domínio "exemplo.com" e para os nomes que a máquina possuir.

Linha 7: Esepecifica as redes confiáveis

Linha 8: Não faz **relay** de redes não confiáveis

Linha 10: Necessário somente quando o mailhost tem redirecionamento para E-Mail não local, via servidor de E-Mails num Firewall.

Os colchetes [] forçam o Postfix a não procurar pela sintaxe MX no DNS.

Nessas condições os usuários acessam seus mailboxes da seguinte maneira:

Acesso ao Mailbox via NFS ou equivalente

Acesso ao Mailbox via POP ou IMAP

Mailbox na máquina preferencial do usuário.

No último caso cada usuário tem um **alias** no mailhost que redireciona os E-Mails para as máquinas preferenciais:

/etc/aliases:

joao: joao@joao.pref.maquina

maria: maria@maria.pref.maquina

Em muitos Unix o **banco de dados de alias** não está em /etc/aliases. Para criá-lo execute o comando:

postconf alias_maps

Execute o comando **"newaliases"** toda vez que alterar o arquivo de **alias.**

Outras configurações podem ser encontradas no URL: http://www.postfix.org/STANDARD_CONF

1.4.3 Utilitários de linha de comando:

postfix inicia e pára o sistema de correio

postalias faz o comando **newaliases** funcionar

postcat lista o conteúdo de arquivos na fila

postconf exibe e edita o arquivo de configuração de correio, **main.cf**

postdrop adiciona mensagens à fila de maildrop

postkick,postlock,postlog fornece bloqueio e registro em **log** para **scripts** de **shell**

postmap cria tabela de base de dados, como o comando **makemap** do Unix

postsuper gerencia as filas (executa na inicialização)

Mais características, funcionalidades e detalhes de configuração do Postfix podem ser obtidas nos seguintes URLs:

http://www.postfix.org
http://www.conectiva.com/doc/livros/online/10.0/servidor/pt_BR/ch11s02.html
www.linorg.cirp.usp.br/Guias.Conectiva/Guias_V.9.0/servidor/correioeletronico.html#POSTFIX

1.5 Testando a Configuração

Para testar a configuração é necessário certificar-se de que as mensagens estão chegando corretamente ao seu servidor e que as mensagens com destino fora do seu servidor estão sendo enviadas corretamente.

O administrador poderá iniciar os seus testes, experimentando enviar uma mensagem a partir do próprio servidor para um outro usuário também localizado no servidor (verifique se o pacote **mailx** está instalado em seu sistema).

Isso pode ser feito através de um terminal, digitando-se na linha de comando:

 # echo teste | mail usuario@minhaorganizacao

LEMBRETE: Após fazer as alterações nos arquivos de configuração, o servidor deve ser reinicializado. Use os comandos a seguir:

$> postfix stop
$> postfix start ou
$> postfix reload

Verificação das configurações :
$> postconf -n

1.6 Exemplo de configuração

O arquivo abaixo (**main.cf**) traz um exemplo de configuração de um servidor Postfix instalado num Debian Linux.

Utilize sempre a versão mais nova. Obtenha-a a partir do site (www.postfix.org)

*** Arquivo /etc/postfix/main.cf:**

```
# See /usr/share/postfix/main.cf.dist for a commented, fuller
    # version of this file.
    # Do not change these directory settings - they are critical to Postfix operation.
    # Diretórios e arquivos componentes do Postfix
    command_directory = /usr/sbin
    daemon_directory = /usr/lib/postfix
    program_directory = /usr/lib/postfix
    smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
    setgid_group = postdrop
    biff = no
    masquerade_domains = $mydomain
    # appending .domain is the MUA's job.
    append_dot_mydomain = no
    # Hostname e domainname de seu servidor
    myhostname = servidor.unidade.usp.br
    mydomain = servidor.unidade.usp.br
    # Lista de aliases de e-mails
    alias_maps = hash:/etc/aliases
    alias_database = hash:/etc/aliases
    myorigin = $mydomain
    mydestination = servidor.unidade.usp.br, localhost.servidor.unidade.usp.br, localhost
    # Quem faz RELAY
    relay_domains = $mydestination, $myhostname, $mynetworks, 143.107.X.0/24, 143. 107.Y.0/23
```

```
relayhost = $mydomain
mynetworks = 143.107.X.0/24, 143.107.Y.0/24, 127.0.0.0/8
mailbox_command = procmail -a "$EXTENSION"
#Tamanho das mensagens
message_size_limit = 10240000
#Tamanho do mailbox
mailbox_size_limit = 20480000
recipient_delimiter = +
#Diretório de fila de e-mails
mail_spool_directory = /var/spool/mail
# Quem envia os E-Mails
mailbox_command = /usr/bin/procmail
# JUNK MAIL CONTROLS
# # The controls listed here are only a very small subset. See the file
# sample-smtpd.cf for an elaborate list of anti-UCE controls.
# The disable_vrfy_command parameter allows you to disable the SMTP
# VRFY command. This stops some techniques used by spammers to harvest
# email addresses.
# disable_vrfy_command = yes
# The smtpd_helo_required parameter optionally turns on the requirement
# that SMTP clients must introduce themselves at the beginning of an
# SMTP session. smtpd_helo_required = yes
# The strict_rfc821_envelopes configuration parameter controls whether
# the Postfix SMTP server requires that MAIL FROM and RCPT TO addresses
# are specified within <>, and that MAIL FROM and RCPT TO addresses
# do not contain RFC822-style comments or phrases. It's great to
# stop SPAM mailers. But it also trips up broken peecee clients.
# # By default, Postfix SMTPD allows RFC822 syntax in MAIL FROM and RCPT TO.
# strict_rfc821_envelopes = yes
# Aqui, a checagem eh feita atraves do HELO, remetente e destinatario
# Bloqueia falsos usuários, falsos remetentes, etc.
smtpd_client_restrictions =
smtpd_helo_restrictions =
smtpd_sender_restrictions =
smtpd_recipient_restrictions =
permit_mynetworks,
 reject_non_fqdn_sender,
 reject_non_fqdn_recipient,
 reject_unknown_sender_domain,
 reject_unknown_recipient_domain,
 reject_unauth_destination,
 reject_unauth_pipelining,
 reject_unknown_client,
 reject_invalid_hostname,
 reject_non_fqdn_hostname,
 permit
# The header_checks parameter restricts what may appear in message
# headers. This requires that POSIX or PCRE regular expression support
# is built-in. Specify "/^header-name: stuff you do not want/ REJECT"
# in the pattern file. Patterns are case-insensitive by default. Note:
# specify only patterns ending in REJECT (reject entire message) or
# IGNORE (silently discard this header). Patterns ending in OK are
# mostly a waste of cycles.
#
# Examina os cabeçalhos e faz filtragem
#header_checks = regexp:/etc/postfix/rejected.he
#header_checks = pcre:/etc/postfix/filename
#
# Aqui, a diretiva body_checks checa por anexos indesejaveis
body_checks = regexp:/etc/postfix/rejected.bo
# SHOW SOFTWARE VERSION OR NOT
#
# The smtpd_banner parameter specifies the text that follows the 220
# code in the SMTP server's greeting banner. Some people like to see
# the mail version advertised. By default, Postfix shows no version.
#
```

```
# You MUST specify $myhostname at the start of the text. That is an
# RFC requirement. Postfix itself does not care.
#
#smtpd_banner = $myhostname ESMTP $mail_name
#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version) (Mandrake Linux)
smtpd_banner = $myhostname NO UCE ESMTP
default_process_limit = 150
qmgr_message_recipient_limit = 40000
```

Alterações nos arquivos **/etc/mail/aliases, dentre outros,** devem ser atualizadas com o comando **postmap.**

Examine os comandos que pertencem ao pacote **Postfix** e veja como eles devem ser usados.

Comandos para controle de filas, de atualização, verificação de **logs**, etc.

Uma cláusula importante que pode ser adicionada ao arquivo **main.cf:**

notify_classes

Essa cláusula determina quais problemas são levados ao conhecimento do **postmaster**. Muitas classes podem resultar em tantas mensagens que ele pode não estar preparado. O padrão é:

notify_classes = resource, software

o que limita os erros a problemas da máquina **host** e a problemas de **software** do **Postfix**.

A tabela a seguir mostra os valores que podem ser incluidos em **notify_classes:**

Classe	Problemas com o correio enviado
bounce	Mensagens que não podem ser entregues
2bounce	Rebatimentos duplos (quando a mensagem de rebatimento também rebate)
delay	Mensagens atrasadas na fila (somente cabeçalho)
policy	Rejeições ao **SPAM** (inclui transcrição de SMTP)
protocol	Erros de protocolo (inclui transcrição de SMTP)
resource	Problemas de recurso (falhas na gravação das filas, sistemas de arquivo cheio, etc)
software	Erros internos "isso não pode ocorrer" do Postfix

A configuração do arquivo **master.cf** é mais complexa. Neste arquivo podem ser colocados itens para que o Postfix trabalhe com Softwares Antivírus, tais como : **Uvscan (da McAfee), Clamav, Dr. Web, Trendmicro, Sophos, etc, e com sistemas intermediários como o Amavis.**

Exemplos de configuração podem ser obtidos nos seguintes URLs:

http://www.securityanalyze.com/gsoares/clamav_postfix.html

http://www.rhbr.com.br/modules.php?name=News&file=article&sid=108

http://www.underlinux.com.br/artigo311.html

Use o pacote WEBMIN para fazer as configurações do Postfix e veja as facilidades/dificuldades que podem ser encontradas.

Instalar o WEBMIN através de pacote encontrado no URL: **http://www.webmin.org**

1.6.1 Cuidados nas configurações

Muitos cuidados devem ser tomados em relação a edição e Endereços de IP e de Rede que vão usar os Postfix, enviar e receber e-mails, fazer RELAY, etc.

As listas **RBL (Relay Black List), SBL (Spamhaus Block List), XBL (Exploits Block List)** e outras devem ser usadas com cautela.

Algumas são de uso comercial e outras são de uso livre.

Veja mais detalhes em **http://www.spamhaus.org**

Sempre tenha cuidado com Relay de E-Mails, faça os testes constantemente e verifique se o seu Postfix está sendo usado para enviar e-mails de outros domínios.

O artigo "Controle de SPAM baseado em pré-detecção da vulnerabilidade de Mail Relay" aborda este assunto:
http://www.rnp.br/newsgen/0207/mail_relay.html

Testes de Relay de E-Mails:
http://www.abuse.net/relay.html

1.6.2 Exemplo do master.cf

O exemplo a seguir mostra parcialmente o arquivo preparado para o **Antivírus CLAMAV:**

```
# DO NOT SHARE THE POSTFIX QUEUE BETWEEN MULTIPLE POSTFIX INSTANCES.
# # ==========================================================
# service type private unpriv chroot wakeup maxproc command + args
# (yes) (yes) (yes) (never) (100)
# ==========================================================
smtp inet n - n - - smtpd
-o content_filter=clamav:clamav clamav unix - n n - - pipe
    flags=Rq user=clamav argv=/usr/lib/postfix/clamav-filter.sh -f ${sender} – ${recipient}
#submission inet n - n - - smtpd
# -o smtpd_etrn_restrictions=reject
#628 inet n - n - - qmqpd ....
....
....
```

1.7 Configurações gerais

1.7.1 Reescrevendo Endereços

http://www.postfix.org/ADDRESS_REWRITING_README.html

O padão de endereços de e-mail (RFC 2822) deve ser obedecido. O Postfix tenta obedecê-lo.

Certos e-mails são reescritos automaticamente. O valor de **mydomain** é apendado ao endereço que possui só o **hostname**, automaticamente.

usuario@servidor torna-se usuario@servidor.exemplo.com.br

Endereços canônicos: Usa o parâmetro **canonical_maps**

- Endereços locais:

joao@exemplo.com.br joao.silva@exemplo.com.br
maria@exemplo.com.br maria.das.dores@exemplo.com.br

- Endereços completos:

pedro@exemplo.com.br pedro.silva@dominio.externo.com
jose@exeplo.com.br jose.maria@uol.com.br

- A relação de endereços de e-mails é armazenada num arquivo externo.

No arquivo **main.cf** deve-se indicar o arquivo dos endereços canônicos:
canonical_maps = hash:/etc/postfix/canonical

Para criar o "arquivo canônico" e atualizar a lista de e-mails para o Postfix:
$> postmap /etc/postfix/canonical
$> postfix reload

- **sender_canonical_maps** : Especifica o canônico do remetente, p. ex:

joao@exemplo.com.br torna-se joao@empresa.com.br, mas o E-Mail é enviado como sendo do usuario@exmplo.com.br

http://www.postfix.org/postconf.5.html#sender_canonical_maps

- **recipient_canonical_maps**: Especifica a tabela para os endereços do cabeçalho e conteúdo da mensagem.

Ambos são processados antes do **canonical_maps.**

1.7.2 Domínios virtuais

O servidor atende a múltiplos domínios na Internet. Para cada domínio virtual há a lista de alias e lista de mailboxes.

Os usuários podem ser do Unix local ou de Banco de dados em rede (LDAP, MySQL ou PostgreSQL).

Os domínios podem ser diferentes e os usuários são do Sistema Unix: **virtual_alias_domains e virtual_alias_maps**

O Mailbox Virtual possui domínios virtuais e usuários não Unix.

Exemplo:

```
1 /etc/postfix/main.cf:
2 virtual_mailbox_domains = example.com ...more domains...
3 virtual_mailbox_base = /var/mail/vhosts
4 virtual_mailbox_maps = hash:/etc/postfix/vmailbox
5 virtual_minimum_uid = 100
6 virtual_uid_maps = static:5000
7 virtual_gid_maps = static:5000
8 virtual_alias_maps = hash:/etc/postfix/virtual
9
10 /etc/postfix/vmailbox:
11 info@example.com example.com/info
12 sales@example.com example.com/sales/
13 # Comment out the entry below to implement a catch-all.
14 # @example.com example.com/catchall
15 ...virtual mailboxes for more domains...
16
17 /etc/postfix/virtual:
18 postmaster@example.com postmaster
```

1.7.3 Outros tipos de configuração

- **Non-Postfix mailbox store: separate domains, non-UNIX accounts:** Atende a múltiplos domínios virtuais e ter as contas em sistemas não Unix.**:**

- **Mail forwarding domains :** Faz o redirecionamento de E-mails sem ter contas de usuários locais. Apenas redireciona os e-mails para outros locais de destino.

- **Mailing lists:** Lista de E-Mails, podendo usar o Majordomo, Mailman ou outro.

Virtual aliases e Virtual mailboxes não podem enviar e-mails através de Softwares de Listas de E-Mails.

A solução é configurar um **virtual aliases** que direciona os endereços virtuais para o agente de envio local:

/etc/postfix/main.cf:

virtual_alias_maps = hash:/etc/postfix/virtual
/etc/postfix/virtual:
listname-request@example.com listname-request
listname@example.com listname
owner-listname@example.com owner-listname
/etc/aliases:
listname: "|/some/where/majordomo/wrapper ..."
owner-listname: ...
listname-request: ...

- **Autoreplies**

Para configurar um **autoreply** em domínios virtuais enquanto está enviando e-mails normalmente, configura-se regras nas tabelas **virtual alias:**
/etc/postfix/main.cf:
virtual_alias_maps = hash:/etc/postfix/virtual
/etc/postfix/virtual:
user@domain.tld user@domain.tld, user@domain.tld@autoreply.mydomain.tld
Isso faz com que os e-mails sejam enviados normalmente e envia uma cópia do e-mail para o endereço que produz um reply automático.

- **http://www.postfix.org/VIRTUAL_README.html**

1.7.4 Autenticação com SASL

A utilização com o Cyrus SASL torna o Postfix um Servidor seguro. O Dovecot também tem a mesma função porém é mais complexo.

A Biblioteca Cyrus SASL apresenta muitos programas e o Postfix pode ser compilado e instalado com estes recursos.

O Postfix SASL possui suporte ao RFC 2554 e pode ser usado para autenticar clientes SMTP remotos ao servidor SMTP. Pode ser usado também para autenticar um cliente SMTP Postfix para um servidor remoto SMTP.

Os logs de **username, métodos de autenticação, e remetente** são colocados no arquivo de **maillog** (geralmente /var/log/maillog) e, como opção, garante acesso aos e-mails via restrição **UCE permit_sasl_authenticated. UCE (Unsolicited Comercial E-Mail) significa E-Mail Comercial Não Solicitado.**

No envio de e-mails o Postfix verifica o hostname do servidor ou o domínio de destino na tabela de senhas do Postfix SASL, e procura por username e senhas. Os dados serão usados para se autenticar ao servidor.

A versão 2.3 do Postfix pode ser configurada para prourar as senhas na tabela SASL pelo endereço de e-mail do remetente.

A compilação e instalação do SASL e do Dovecot devem ser efetuadas observando a versão do Postfix. A habilitação do recurso de SASL, utilização e configuração estão descritos na URL : **http://www.postfix.org/SASL_README.html**

1.7.4.1 Habilitando o SASL com um Servidor SMTP

- Para estabelecer um servidor SMTP com suporte a SASL, deve-se fazer basicamente:

/etc/postfix/main.cf:
smtpd_sasl_auth_enable = yes
In order to allow mail relaying by authenticated clients:
/etc/postfix/main.cf:
smtpd_recipient_restrictions =
permit_mynetworks permit_sasl_authenticated ...

- Para reportar os login names do SASL login names nos e-mails recebidos: message headers (Postfix version 2.3 and later):

/etc/postfix/main.cf:
smtpd_sasl_authenticated_header = yes

- Nota: Os login named do SASL serão compartilhados com todos.

Versões antigas de Softwares clientes SMTP da Microsoft implementam uma versão não padronizada das sintaxe do protocolo AUTH e espera que o servidor SMT responda **EHLO** com **"250 AUTH=stuff"** ao invés de **"250 AUTH stuff"**. Para satisfazer estes clientes use o seguinte:
/etc/postfix/main.cf:
broken_sasl_auth_clients = yes

1.7.4.2 Configuração DOVECOT SASL para o SMTP Postfix

Disponível para as versões 2.3 e superiores. Basta apenas especificar a localização do daemon socket da autenticação Dovecot. O que se usa gerlamente é o caminho relativo ao diretório das filas de e-mail do Postfix (**queue**), que vai trabalhar juntamente com o Postfix (em modo **chroot** ou não)
/etc/postfix/main.cf:
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
Em relação ao Dovecot será necessário especificar o caminho absoluto do daemon socket de autenticação. Assumindo que o caminho da **queue** do Postfix seja **/var/spool/postfix/**
/some/where/dovecot.conf:
auth default {
mechanisms = plain login
passdb pam {
}
userdb passwd {
}
socket listen {
client {
path = /var/spool/postfix/private/auth
mode = 0660
user = postfix
group = postfix
}
}
}
See the Dovecot documentation for how to configure and operate the Dovecot authentication server.

1.7.4.3 Configuração do Cyrus SASL para o servidor SMTP Postfix

- Para autenticar usando o banco de dados de senhas do Unix:

(Cyrus SASL versão 1.5.5)
/usr/local/lib/sasl/smtpd.conf:
pwcheck_method: pwcheck
(Cyrus SASL versão 2.1.1)
/usr/local/lib/sasl2/smtpd.conf:
pwcheck_method: pwcheck

O arquivo de configuração (/usr/local/lib/sasl - Cyrus SASL version 1.5.5) ou /usr/local/lib/sasl2 - Cyrus SASL version 2.1.1) é usado pela biblioteca SASL pode ser usado com:
/etc/postfix/main.cf:
smtpd_sasl_application_name = smtpd (Postfix < 2.3)
smtpd_sasl_path = smtpd (Postfix 2.3 and later)

- Para autenticar com o arquivo de senhas do Cyrus SASL:

(Cyrus SASL version 1.5.5)
/usr/local/lib/sasl/smtpd.conf:
pwcheck_method: sasldb
(Cyrus SASL version 2.1.1)
/usr/local/lib/sasl2/smtpd.conf:
pwcheck_method: auxprop
(Cyrus SASL version 1.5.5)

- Criação da base de dados dos usuários:

% saslpasswd -c -u 'postconf -h myhostname' usuarioexemplo
(Cyrus SASL version 2.1.1)
% saslpasswd2 -c -u 'postconf -h myhostname' usuario exemplo
O comando **sasldblistusers** (Cyrus SASL version 1.5.5) ou **sasldblistusers2** (Cyrus SASL version 2.1.1) permite listar e administrar os usuários cadastrados.
No lado dos servidor Postfix todos os usuários devem ser cadastrados.
A variável **smtpd_sasl_local_domain** controla as autenticações usadas pelo **daemon smtpd:**
/etc/postfix/main.cf:
smtpd_sasl_local_domain = $myhostname

- Importante: Todos usuários podem ser autenticados usando **todos** os mecanismos de autenticação oferecidos pelo Postfix e a autenticação deve terminar (ou falhar) com mecanismos não suportados. Se for usado o SASL para autenticar (usando o **saslauthd**), ao invés de usar o PAM (pluggable authentication modules), somente os mecanismos de login e senhas legíveis (plain passwords) são suportados. A biblioteca SASL pode pedir outros mecanismos, tais como o DIGEST-MD5. Isso acontece porque estes mecanismos são utilizados por outros plugins e a biblioteca SASL não tem como saber que a sua única fonte de autenticação é o PAM**. Assim, é necessário limitar a lista de mecanismos citadas pelo Postfix.**

- Em versões velhas do Cyrus SASL deve-se remover os arquivos de bibliotecas correspondentes e fazer as devidas atualizações.

- Com Cyrus SASL versão 2.1.1 ou posterior:

/usr/local/lib/sasl2/smtpd.conf:
mech_list: plain login

- Com Cyrus SASL versão 1.5.5 você deve apagar o diretório das bibliotecas correspondentes ao plug-in do SASL

Com o SASL versão 2.1.1:
/usr/local/lib/sasl2/smtpd.conf:
pwcheck_method: auxprop
auxprop_plugin: sql

1.7.4.4 Testando a autenticação SASL no servidor SMTP Postfix

Conecte-se ao servidor, porta SMTP (25), e veja a tela de diálogo. As informações enviadas pelo cliente estão em negrito.

220 server.example.com ESMTP Postfix
EHLO client.example.com 250-server.example.com
250-PIPELINING
250-SIZE 10240000
250-ETRN
250-AUTH DIGEST-MD5 PLAIN CRAM-MD5
250 8BITMIME
AUTH PLAIN dGVzdAB0ZXN0AHRlc3RwYXNz
235 Authentication successful

Ao invés de **dGVzdAB0ZXN0AHRlc3RwYXNz**, especifique a forma codificada de username\0username\0password (\0 é um byte nulo) codificada em 64 bits

O exemplo anterior é para um usuário 'test' com senha 'testpass'.

Para gerar uma informação autenticada em codificação base 64 use um dos seguintes comandos:

$> % printf 'username\0username\0password' | mmencode
$> % perl -MMIME::Base64 -e \ 'print encode_base64("username\0username\0password");'

O comando **mmencode** é parte do metamail software **MIME::Base64**.

Está disponível em **http://www.cpan.org**

- **Cuidados**: Se os logs das negociações SASL estiverem em listas públicas, tenha em mente que é bastante trivial recuperar o username/password das informações codificadas em Base64.

1.7.4.5 Habilitando o SASL no Cliente SMTP Postfix

- Habilite no lado do cliente a autenticação SASL e especifique a tabela com informações de username e senhas por host ou por destino. O Postfix procura, inicialmente, entradas com server hostname; e se não encontrar então procura entradas com **next-hop destination**. Geralmente o lado direito da tabela traz os endereços de E-Mail, mas pode ser informações especificadas com os parâmetros de relayhost ou tabelas de transporte.

/etc/postfix/main.cf:
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_type = cyrus
/etc/postfix/sasl_passwd:
foo.com username:password
bar.com username
[mail.myisp.net] username:password
[mail.myisp.net]:submission username:password

- O Postfix versão 2.3 suporta informações de senhas de SASL por remetente. Para procurar as senhas por usuário antes de chegar ao destino, especifique:

/etc/postfix/main.cf:
smtp_sender_dependent_authentication = yes
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
/etc/postfix/sasl_passwd:
user@example.com username:password
bar.com username
[mail.myisp.net] username:password
[mail.myisp.net]:submission username:password

- Nota: Alguns servidores SMTP suportam senhas planas ou autenticação de LOGIN somente. Por padrão o Postfix não usa métodos de autenticação que envia senhas em texto calro (**plaintext passwords**) e as envia com a seguinte mensagem de erro:

"Authentication failed: cannot SASL authenticate to server".

- Para habilitar a autenticação de senhas **plaintext**, especifique o seguinte:

/etc/postfix/main.cf:
smtp_sasl_security_options = noanonymous

- A senha do cliente SASL Postfix é aberta antes do servidor SMTP entrar no modo **chroot jail** (onde você pode deixá-lo em /etc/postfix).

- Nota: Alguns servidores SMTP suportam mecanismos de autenticação que, disponíveis no cliente, podem não trabalhar na prática, ou possuir credenciais apropriadas para se autenticar ao servidor. Isto é possível através do parâmetro **smtp_sasl_mechanism** para restringir futuramente a lista de mecanismos do servidor que o cliente **smtp** vai considerar :

/etc/postfix/main.cf: smtp_sasl_mechanism_filter = !gssapi, !external, static:all

- No exemplo anterior o Postfix vai aceitar o uso de macanismos que requerem uma estrutura especial, tal como o **Kerberos**.

- O Postfix é compatível com clientes que não usam a sintaxe de resposta não padronizada **"AUTH= method...."** na resposta ao comando **EHLO.**

1.8 Encriptação com TLS

Transport Layer Security (TLS, geralmente chamado de SSL) provê autenticação baseada em certificados e sessões criptografadas.

Uma sessão criptografada protege a informação transmitida com **SMTP Mail** ou com autenticação **SASL**.

http://www.postfix.org/TLS_README.html

1.9 Controle de accessos e de RELAY

1.9.1 Controle de RELAY e controle de acessos

Relay control, junk mail control, and per-user policies Permite controlar o fluxo de e-mails no servidor, tentando evitar o abuso de SPAMs e grande fluxo de e-mails.

O ideal é permitir o tráfego de e-mails apenas das redes autorizadas e dos usuários cadastrados.

Os controles de Relay trabalham em cima de endereços de Rede, de IPs, interfaces de Rede, domínios virtuais, **alias** de domínios e interfaces de proxy.

O principal objetivo é o controle de acesso de forma a evitar o grande fluxo de e-mails **(junk emails).**

1. **Orientação a protocolo** : Bloqueios de características do SMTP, Softwares para fazer **junk emails**, tráfego de Worms com implementações de clientes SMTP não padronizadas. Esse controle é pouco usado à medida que os Spammers e os criadores de Worms aprendem a ler as documentações RFC.

2. **Orientação a Blacklist**: Alguns servidores SMTP fazem o controle baseado em buscas de blacklists que relacionam sites mal configurados com Open Mail Relays, Open Web Proxies e computadores Desktop comprometidos e sob controle remoto de criminosos e Software maliciosos. A eficiência das **blacklists** depende de quanto completas e atualizadas elas estão.

3. **Orientação a comparações**. Controle baseado em listas **grey** (**greylisting**) ou consultando os endereços de remetente e destinatário, através do **SPF**.

Estes recursos são implementados externamente e são assuntos do item documento **:**
SMTPD_POLICY_README.
A verificação de **endereços de Sender/recipient** é o assunto do documento **:**
ADDRESS_VERIFICATION_README

1.9.2 Restrições que se aplicam a todo SMTP

Por cliente e por usuário:

- header_checks e body_checks, aplicadas aos conteúdos dos e-mails antes de entrar nas filas de entrada.
- Exigência de HELO ou EHLO antes de enviar os comandos MAIL FROM ou ETRN, podem ser causados por aplicações que enviam e-mails. O default é desabilitado.
- Desabilitar a sintaxe ilegad nos comandos MAIL FROM ou RCPT TO. Isso pode causar problemas com aplicações que enviam e-mails e com clientes de e-mails em computadores Desktop velhos demais. O padrão é estar desabilitado**: "strict_rfc821_envelopes = no".**

Exemplos:

1. **Disallowing RFC 822 address syntax (example: "MAIL FROM: the dude <dude@example.com>"**
2. **Disallowing addresses that are not enclosed with <> (example: "MAIL FROM: dude@example.com").**

- Rejeitar e-mails de endereços de remetente não existentes. Esse tipo de **"filtro"** ajuda a acabar com os **worms e malwares,** mas pode causar problemas com aplicativos que enviam e-mails com endereços sem **reply.** Por esta razão o padrão é estar desabilitado : **"smtpd_reject_unlisted_sender = no"**
- Rejeitar e-mails de endereços destino não existentes. Isso ajuda a fila de e-mail ficar livre de mensagens do tipo **undeliverable MAILER-DAEMON**. É habilitado por default : **"smtpd_reject_unlisted_recipient = yes"**

1.9.3 Restringindo o SMTP com listas de acesso restritas

Permite especificar a lista de restrição de acesso para cada estágio da conversação SMTP
/etc/postfix/main.cf:

```
# Allow connections from trusted networks only.
smtpd_client_restrictions = permit_mynetworks, reject

# Don't talk to mail systems that don't know their own hostname.
# With Postfix < 2.3, specify reject_unknown_hostname.
smtpd_helo_restrictions = reject_unknown_helo_hostname

# Don't accept mail from domains that don't exist.
smtpd_sender_restrictions = reject_unknown_sender_domain

# Whitelisting: local clients may specify any destination. Others may not.
smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination
```

```
# Block clients that speak too early.
smtpd_data_restrictions = reject_unauth_pipelining

# Enforce mail volume quota via policy service callouts.
smtpd_end_of_data_restrictions = check_policy_service unix:private/policy
```

1.9.4 Delayed evaluation of SMTP access restriction lists

1.9.5 Dangerous use of smtpd_recipient_restrictions

1.9.6 SMTP access rule testing

http://www.postfix.org/SMTPD_ACCESS_README.html

1.10 Policiamento de acessos

http://www.postfix.org/SMTPD_POLICY_README.html

1.11 Verificação de endereços

http://www.postfix.org/ADDRESS_VERIFICATION_README.html

1.12 Controle de acessos

http://www.postfix.org/RESTRICTION_CLASS_README.html

1.13 ETRN

http://www.postfix.org/ETRN_README.html

1.14 Postfix e UUCP

http://www.postfix.org/UUCP_README.html

1.15 Análise de Logs do Postfix

Softwares para interpretar os logs do Postfix. Geralmente relacionam o número de mensagens de E/S (por usuários e totalizadas), erros, tamanho das mensagens, alertas, mensagens bloqueadas por Listas Anti-SPAM, etc.

Alguns exemplos de Softwares:

PF-Graph, pflogsumm, isoqlog, mailgraph, etc.

1.16 Controle de SPAM

O **Postfix** utiliza expressões regulares, tabelas de base de dados e as listas negras do projeto **MAPS** para filtrar **SPAM**. A tabela a seguir mostra algumas das variáveis do **Postfix** relacionado ao **SPAM.**

Se uma mensagem corresponder a uma pesquisa em uma tabela e o valor de tabela for **REJECT**, a mensagem será rejeitada com uma mensagem de erro adequada. Para os **hackers** do

Perl, eis um exemplo de expressão regular utilizada em filtros de **SPAM** de um **site**:
/^friend@.$/ 550 Stick this in your pipe $0
Se realmente tiver um usuário chamado **"friend"** em seu domínio, você poderia excluir esse usuário da mensagem amigável de erro com:
/^friend@(?!meusite.com).*$/ 550 Stick this in your pipe $0

Variável	Significado
header_checks	Filtros em cabeçalho
smtpd_client_restrictions	Filtros em conexões de cliente, listas negras, etc
smtpd_sender_restrictions	Filtros nos endereços do remetente
smtpd_recipient_restrictions	Filtros nos endereços do destinatário
smtpd_helo_required	Exige SMTP HELO com nome de **host** identificado
smtpd_helo_restrictions	Exige pesquisa inversa de DNS
smtpd_etrn_restrictions	Lista **hosts** com permissão para solicitar execuções de fila

Para utilizar as **listas negras** do projeto **MAPS**, adicione o seguinte ao seu arquivo **main.cf**:
maps_rbl_domains =opm.blitzed.org,
list.dsbl.org,
blackholes.easynet.nl,
zombie.dnsbl.sborbs.net,
bl.spamcop.net,
sbl.spamhaus.org
rbl.maps.vix.com
dul.maps.vix.com
relays.mail-abuse.org
smtpd_client_restrictions = reject_maps_rbl

Atenção para as modificações futuras:
warning: support for restriction "reject_maps_rbl" will be removed from Postfix; use "reject_rbl_client domain-name" instead

- **Consulte estas listas antes para ver os e-mails, domínios, endereços IPs, etc, cadastrados. Há listas que rejeitam todos os e-mails originados de certas regiões e de Redes Classe C completas.**

Capítulo 2

Postfix com Antivírus

Nos dias de hoje é comum receber uma infinidade de **e-mails** com vírus. Os servidores de **E-Mails** tem que entregar as mensagens.

Além de sua função básica, entregar mensagens, agora ele tem que tratar os **e-mails**, verificar o que é SPAM, o que é vírus e aplicar os filtros necessários. Clientes de Webmail, tais como o **Openwebmail** já estão com filtros prontos para utilizar.

Muitos tutoriais ensinam como instalar o **postfix + amavis + clamav, postfix + clamav, etc.**

O **AMAVIS** é uma interface que atua entre o servidor de **E-Mails** e o **Antivírus.** O site do projeto é **http://www.amavis.org**

O **CLAMAV** é um sistema de Antivírus para Software Livre (**www.clamav.net**)

O **UVSCAN** é um sistema de Antivírus da McAfee (**www.mcafee.com**)

O **SOPHOS** é um sistema de Antivírus e Anti-SPAM da Sophos (**www.sophos.com**)

Para instalar esse conjunto todo, deve-se observar os Softwares exigidos, suas versões, e obedecer todos os itens de configuração e detalhes envolvidos.

2.1 Instalação do Clamav

Os pacotes relacionados ao **clamav** para o Debian podem ser obtidos pelo comando :

$> apt-cache search clamav
clamav - antivirus scanner for Unix
clamav-base - base package for clamav, an anti-virus utility for Unix
clamav-daemon - antivirus scanner daemon
clamav-docs - documentation package for clamav, an anti-virus utility for Unix
clamav-freshclam - downloads clamav virus databases from the Internet
clamav-milter - antivirus scanner for sendmail
clamav-testfiles - use these files to test that your Antivirus program works
libclamav-dev - clam Antivirus library development files
libclamav1 - virus scanner library
sylpheed-claws-clamav - Clam AntiVirus plugin for Sylpheed Claws
clamav-data - clamav data files
clamav-getfiles - Update script for clamav
clamcour - courier filter for clamav to virus scan incoming mail
clamsmtp - virus-scanning SMTP proxy
courier-filter-perl - purely Perl-based mail filter framework for the Courier MTA

- **$> apt-get install clamav**
- **$> apt-get install clamav-daemon**

- O **usuário clamav** e o **grupo clamav** deverão ser estabelecidos após a instalação dos pacotes.

O **CLAMAV** pode ser instalado e trabalhar nos modos:

Daemon : O **daemon** permanece ativo sempre e é aconselhável para conxeões permanetntes de rede.

ifup.d : Neste modo o **freshclam** será executado como **daemon** assim que a conexão de rede é inicializada. Bom utilizar em conexões Dial-UP.

cron :O **freshclam** é inicializado pelo **cron.** O controle de atualização do banco de dados é feito pelo **freshclam.**

manual : A chamada do freshclam não inicia automaticamente. Não recomendado pois não se tem uma atualização constante do banco de dados

Depois de instalado, um banco de dados relacionando os vírus e suas assinaturas deverá ser instalado de um **mirror**.

A detecção de vírus é feita pelo executável **/usr/bin/clamscan**

- A utilização do **clamscan** é simples, basta executar alguns comandos:

$> clamscan /home/usuario/*

$> clamscan arquivo

Há um arquivo de teste para antivírus que contém 68 bytes de tamanho. Pode ser obtido do URL:

http://www.eicar.org/download/eicar.com

Faça o **download** deste arquivo e teste o **clamscan. O conteúdo deste arquivo é :**

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

- O comando **sigtool** serve para manipular o banco de dados de assinaturas de vírus:

$> sigtool -l irá mostrar todas as assinaturas de vírus no Banco de dados.

2.2 Exemplos de configuração

- O exemplo mostrado em **/usr/share/doc/clamav/examples/clamavmon** tem por finalidade monitorar servidores **clamav** rodando numa rede. Se houver falhas nestes servidores **Clamav** um alerta será emitido.
- O exemplo mostrado em **/usr/share/clamav/examples/clamdmon** possui traz o programa **ClamdMon** que vai monitorar o **daemon do clamav.** Um padrão de assinatura (o arquivo de teste de assinatura eicar.com) é enviado ao daemon do clamav. Se for encontrado um vírus será retornado o valor UM, caso contrário ZERO.
- O exemplo mostrado em **/usr/share/clamav/example/clamdwatch** faz a verificação do **daemon do clamav** também, só que gerenciaod pelo **cron**.
- O exemplo **/usr/share/clamav/example/clamdpipe** pode ser usado com o **procmail** para verificar **e-mails** com vírus. Se for encontrado vírus nos e-mails o cabeçalho apresentará a sintaxe **X-Virii-Status.**

2.3 Configurações básicas

O pacote **clamav-base** traz as configurações básicas do **Clamav.**

O diretório /etc/clamav traz arquivos fundamentais para o **clamav** e devem ser editados com muito cuidado e atenção.

2.3.1 Arquivo clamd.conf:

#Automatically Generated by clamav-base postinst
#To reconfigure clamd run #dpkg-reconfigure clamav-base
#Please read /usr/share/doc/clamav-base/README.Debian.gz for details
LocalSocket /var/run/clamav/clamd.ctl
FixStaleSocket
User clamav
AllowSupplementaryGroups ScanMail
ScanArchive
ArchiveMaxRecursion 5
ArchiveMaxFiles 1000
ArchiveMaxFileSize 10M
ArchiveMaxCompressionRatio 250
ReadTimeout 180
MaxThreads 12
MaxConnectionQueueLength 15
LogFile /var/log/clamav/clamav.log
LogTime
LogFileMaxSize 0
PidFile /var/run/clamav/clamd.pid
DatabaseDirectory /var/lib/clamav
SelfCheck 3600
ScanOLE2
ScanPE
DetectBrokenExecutables
ScanHTML
ArchiveBlockMax
Mail files
Enable internal e-mail scanner.
Default: enabled
ScanMail
If an email contains URLs ClamAV can download and scan them.
WARNING: This option may open your system to a DoS attack.
Never use it on loaded servers.
Default: disabled
#MailFollowURLs

2.3.2 Arquivo freshclam.conf

Automatically created by the clamav-freshclam postinst
Comments will get lost when you reconfigure the clamav-freshclam package
DatabaseOwner clamav
UpdateLogFile /var/log/clamav/freshclam.log
LogFileMaxSize 0
MaxAttempts 5
Check for new database 24 times a day
Checks 24
DatabaseMirror db.local.clamav.net
DatabaseMirror database.clamav.net
DatabaseDirectory /var/lib/clamav/
NotifyClamd
DNSDatabaseInfo current.cvd.clamav.net

2.4 Utilização com Postfix

A configuração é bem simples. Basta alterar o arquivo **/etc/postfix/master.cf** e colocar os trechos abaixo:

```
smtp inet n - n - - smtpd
-o content_filter=clamav:clamav clamav unix - n n - - pipe
    flags=Rq user=clamav argv=/usr/lib/postfix/clamav-filter.sh -f ${sender} – ${recipient}
#submission inet n - n - - smtpd
# -o smtpd_etrn_restrictions=reject
#628 inet n - n - - qmqpd ....
```

O script **/usr/lib/postfix/clamav-filter.sh** vai processar os **e-mails** de forma a verificar a existência de vírus**:**

```
#!/bin/sh
# ClamAV script; set a opcao ScanMail no clamav.conf
# by Deives Michellis "thefallen" - dmichellis@yahoo.com | thefallen@unitednerds.org
# Crie uma linha no master.cf com o formato:
#clamav unix - n n - - pipe
# flags=Rq user=clamav argv=/usr/libexec/postfix/clamav-filter.sh -f ${sender} – ${recipient}
# e edite a linha do SMTP assim:
# smtp inet n - n - - smtpd
# -o content_filter=clamav:clamav
export PATH=/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/games
# Config
INSPECT_DIR=/var/spool/filter
SENDMAIL="/usr/sbin/sendmail -i "
MYHOSTNAME=‘postconf -h myhostname‘
REPORTHOST=‘postconf -h myhostname‘
# Exit codes <sysexits.h>
EX_TEMPFAIL=75
EX_UNAVAILABLE=69
EX_DENIED=77
# Definicoes dos nomes temporarios
nome_arquivo=‘date +%Y%m%d%H%M%S‘
nome_arquivo=in.$$.$nome_arquivo
AVCMD="/usr/local/bin/clamdscan –disable-summary –stdout "
NOTIFY_VIRUS=yes
NOTIFY_POSTMASTER=yes
viruscan() {
VIRUS=‘$AVCMD $nome_arquivo‘
SAIDA=$?
VIRUS=‘echo $VIRUS | cut -d" " -f2-‘
if [ $SAIDA -eq 1 ]; then
postlog -t postfix/virus-filter message-id=$msgid reject: VIRUS from=\<$from\> to=\<$rcpts\>
2>/dev/null
if [ "$NOTIFY_VIRUS" = "yes" ]; then
echo "From: Virus Scanner <mailer-daemon@$MYHOSTNAME>
Subject: AVISO: Email rejeitado: VIRUS Detectado
To: $from
Seu email para ($rcpts) com assunto ($subj) foi rejeitado por conter virus.
Virus encontrados: $VIRUS
" | $SENDMAIL -f MAILER-DAEMON – $from
fi
if [ "$NOTIFY_POSTMASTER" = "yes" ]; then
echo "From: Virus Scanner <mailer-daemon@$MYHOSTNAME>
Subject: Postmaster Copy: VIRUS Detectado
To: postmaster@$MYHOSTNAME
Um email de $from para ($rcpts) com assunto ($subj) foi rejeitado por conter virus.
Virus encontrados: $VIRUS
```

```
" | $SENDMAIL -f MAILER-DAEMON – postmaster@$MYHOSTNAME
fi
exit 0
fi
}
#
# Clean up when done or when aborting.
#trap "rm -rf $nome_arquivo*" 0 1 2 3 15
## Start processing.
#cd $INSPECT_DIR || { echo $INSPECT_DIR does not exist; exit $EX_TEMPFAIL; }
cat >$nome_arquivo || { echo Cannot save mail to file; exit $EX_TEMPFAIL; }
from=$2
if [ "$from" != "–" ]; then
shift
else
$from=""
fi
shift ; shift
dominio=`echo $from | cut -d"@" -f2`
email=`echo $from | cut -d"@" -f1`
subj=`head -n 200 $nome_arquivo | grep -i "^Subject:" | cut -d":" -f2- | head -n 1`
msgid=`head -n 200 $nome_arquivo | grep -i "^message-id" | cut -d: -f 2- | sed 's/^ *//' | head -n 1`
saida="-f $from – $@"
rcpts=$@
viruscan
$SENDMAIL $saida <$nome_arquivo
exit 0
```

- Todo e-mail de infectado será enviado ao **postmaster** e rejeitado para entrega ao destinatário.
- É importante manter o freshclam sempre atualizado, com as listas de assinaturas atualizadas.
- Exemplo de instalação para o FreeBSD:

http://granito2.cirp.usp.br/FreeBSD2/clamav.html

- Exemplo de configurações para Linux Debian. Postfix + Clamav + SASL

http://granito2.cirp.usp.br/Debian/clamav/

2.5 Clamav-milter

É uma interface entre o MTA e o antivírus **Clamav. Milter** pode ser denominado como **Mail Filter, ou seja, Filtro de E-Mails.**

Faz a verificação de vírus com o **Clamav** e interage com o MTA. Inicialmente projetado para o Sendmail, permite que trabalhe em ambiente de Rede que possui mais de um Sistema de Anti-Vírus. Tem suporte a **tcpwrappers** e pode ser administrado pelos arquivos **/etc/hosts.allow e /etc/hosts.deny.** A URL abaixo mostra os detalhes e cuidados para a configuração do Postfix com o **Clamav-milter**, utilizando o **S.O. Linux Slackware.**

http://wiki.linuxquestions.org/wiki/Postfix_with_clamav-milter

Há detalhes em permissões de arquivos e diretórios que devem ser seguidas à risca, tais como:

- **$> mkdir /var/run/clamav**
- **$> chown clamav:postfix /var/run/clamav**
- **$> chmod 750 /var/run/clamav**
- **$> mkdir /var/run/clamav/quarantine**
- **$> chown clamav:clamav /var/run/clamav/quarantine**
- **$> chmod 700 /var/run/clamav/quarantine**
- O diretório /**var/run/clamav**/ armazena em arquivo o **PID (Process ID)** e o **socket** do **clamav.**
- O usuário **postfix precisa acessar o milter socket.** O primeiro passo é fazer o diretório /**var/run/clamav**/ ter o grupo **postfix** como dono. O diretório /**var/run/clamav/quarantine** vai armazenar os **e-mails** infectados.
- Para a configuração do Postfix deve-se colocar no arquivo **main.cf:**

smtpd_milters = unix:/var/run/clamav/clamav-milter
milter_default_action = accept

A segunda linha instrui o Postfix a fazer um **bypass (desvio)** do **milter,** se ele falhar, ao invés de reportar um código de erro temporário. Essa linha pode ser removida depois de tudo estar trabalhando como o esperado.

2.6 Amavisd-new

É uma interface entre o MTA (Servidor de E-Mails) e o **scanner de vírus/filtro de conteúdos.**

É mais complexo que a anterior, **clamav-milter.**

Pode trabalhar com AntiSpam, diferentes sistemas de **antivírus** e diferentes servidores de E-Mail. Os e-mails infectados são colocados em quarentena, ou descartados, e o **postmaster** é notificado. Também permite que se especifique o SpamAssassin para tratamento de SPAMS e outros filtros.

A instalação é simples:

$> apt-get install amavisd-new-milter

O **usuário amavis** e o **grupo amavis** deverão ser estabelecidos após a instalação dos pacotes.

As configurações estão no arquivo /**etc/amavis/amavisd.conf** e está dividida em 8 seções:

Seção I: Configurações essenciais e do MTA
Seção II: Específico para cada MTA
Seção III: Logs e armazenamento de logs
Seção IV: Notificações, Rejeição/Descarte/liberação de destinatários, quarentenas, etc.
Seção V: Manipulação por usuários, por recipientes, whitelist, etc.
Seção VI: Limites de recursos
Seção VII: Interação com programas externos, antivírus, SpamAssassin, etc.
Seção VIII: **Debug** e tutoriais

2.7 Referências e Tutoriais

http://www.superphp.com.br/tutoriais/index.php?id=57
http://www.securityanalyze.com/gsoares/clamav_postfix.html
http://www.underlinux.com.br/modules.php?name=Sections&op=printpage&artid=311
http://www.nerdgroup.org/cooltrick/uvscan_postfix.txt
http://www.linuxit.com.br/article60.html

Capítulo 3

Anti SPAM

SPAM é um constante na Internet. É democrático pois atinge a todos os que possuem e-mail e usam a Internet.

No **URL http://www.antispam.br** pode-se encontrar a história do SPAM, conhecer suas origens e curiosidades.

SPAM incomoda tanto que existe hoje diversas técnicas e Softwares, das mais diversas complexidades, empenhados em reduzir o número de SPAMs que sobrecarregam a Internet. Eliminar totalmente o SPAM é como dizer que deve ter uma lei que proíba propaganda. Totalmente impossível pensar que isso um dia pode acontecer.

A convivência com o problema é fatal e, quanto menor for o número de SPAMs recebidos pode-se ter certeza de que há muita complexidade e técnicas de programação envolvidos.

Há muitas técnicas de filtragem de E-mails : por conteúdo, filtros de cabeçalhos, de remetentes, etc.

Amplas estruturas de Redes de Dados podem ser comprometidas pela existência de um Vírus na rede fazendo SPAM. É comum ouvir isso.

3.1 Filtragem de E-Mails

A filtragem é feita no nível de servidor e no nível de usuário. Alguns dos principais filtros são: Filtros Bayesianos, filtros por cabeçalho, filtros por regras sintáticas, filtros por listas e filtro de vírus.

3.1.1 Filtros por cabeçalho:

- Examinam as informações do cabeçalho das mensagens e aplicam regras inseridas no Servidor de E-Mails. Este tipo de filtro é implementado diretamente no **MTA (Mail Transport Agent).** As informações verificadas por este tipo de filtro são:

- **Listas externas:** São listas que possuem endereços de servidores cadastrados. Se o servidor que está enviando uma mensagem estiver nessa lista a mensagem será recusada. Essas listas são chamadas de **Open Relay Databases.** São compostas de servidores que permitem o envio de mensagens com qualquer endereço de origem e a partir de qualquer lugar/rede. Servidores mail configurados permitem isso.
  Faça os testes de Open Relay através do URL : **http://www.abuse.net/relay.html**

- **Lista externa de acesso:** é uma lista onde os usuários podem inserir ou remover um endereço de servidor.

- **Checagem do endereço IP do emissor:** é feita a verificação de DNS reverso e direto, ou seja, se existe um nome para o endereço IP do servidor que está enviando a mensagem. O

nome deverá ser um servidor de E-Mails cadastrado como tal na rede do proprietário. Diz-se que deve ser um **SMTP com MX oficial da empresa.**

- A seguir são mostradas 6 listas de consultas externas, uma lista interna e algumas regras presentes no Postfix:

1. Permitir e-mails enviados a partir de máquinas da própria rede
2. Verificar lista interna de servidores aceitos
3. Verificar lista interna de servidores rejeitados
4. Verificar listas externas
5. Verificar se o domínio do servidor existe
6. Verificar se o cabeçalho está bem composto, de acordo com a norma
7. Verificar se existe o login do emissor
8. Verificar se existe um nome para o endereço IP do servidor emissor.

- Cada usuário pode escolher entre dois níveis de filtragem: nível alto e nível baixo. O nível alto só aceita mensagens que passaram por todos os testes executados pelo **MTA,** citados acima. O nível baixo aceita mensagens enviadas, verificando somente as regras de 1 a 4 .

3.1.2 Filtros de Vírus

- São filtros que recebem o **E-mail** do **MTA** e chamam um **Anti-vírus** para verificar se existe vírus naquele e-mail. Se todos os arquivos anexos da mensagem não possuirem vírus, o filtro retorna a mensagem ao **MTA** para que ela possa ser entregue ao usuário.
  Ultimamente as mensagens de e-mails trazem **links** indicando onde pegar arquivos executáveis, screensavers, etc, o que pode conter vírus, **spyware, spybots, trojans, malwares,** etc.

3.2 Filtros por regras estáticas

- São filtros que verificam se determinado **E-mail** pertence a uma regra sintática pré-definida. Esses filtros podem analisar tanto o cabeçalho quanto o corpo da mensagem.
  Um grande problema desse filtro é que o usuário tem que inserir cada regra manualmente. Esse filtro não tem precisão, uma regra sintática pode bloquear e-mails legítimos. Por exemplo, o usuário cria uma regra que bloqueia todos os e-mails que contiverem a palavra **"sexo"** e vierem de algum endereço ***.ru** . E-Mails vindos da Rússia e que contenham a palavra **sexo** não serão repassados aos usuários finais.

3.3 Filtros Bayesianos

- Classificam as mensagens através da análise de seu conteúdo. O conteúdo da mensagem é comparado com duas bases de dados que armazenam palavras consideradas boas **(goodlist)** e ruins **(spamlist)**. O somatório dos pesos dessas palavras ponderado pela taxa de controle indica a classificação do **E-Mail**

- O peso das palavras é atribuído durante o **treinamento** do filtro. Cada vez que o filtro é treinado o peso de cada palavra é modificado considerando o seu peso atual e o número de vezes que a palavra apareceu no treinamento (a frequência da palavra).

- A taxa de compressão é configurada pelo usuário e indica o percentual que a mensagem precisa atingir para ser classificada como **SPAM.**
- Para uma taxa de controle de 50% a mensagem será classificada como **spam** se o somatório dos pesos das palavras da **base ruim** for maior do que o somatório dos pesos da **base boa.**
- O **Bayesian Mail Filter** é um filtro individual instalado opcionalmente na conta de cada usuário. Cada usuário cria sua base de dados com palavras boas e ruins. Deve-se efutuar um **treinamento** prévio deste filtro. A performance varia de acordo com a implementação. Para funcionar corretamente :

1. O filtro precisa de exemplos de **SPAM** e de **não SPAM, e precisa de mensagens recentes (filtros probabilísticos deven constantemente – ou periodicamente – treinados**
2. Usar a base de dados de outra pesso pod enão funcionar muito bem: se o conteúdo das mensagens for diferente a idéia do filtro não tem muita relevância.
3. As mensagens devem ser do mesmo período.

Algumas das implementações que utilizam este modelo de filtro são:

- **Bayesian Mail Filter**: Projetado para ser pequeno, rápido e mais versátil do que outros **filtros bayesianos.**
- **Bogofilter**: Desenvolvido para ser rápido. Em geral muito utilizado por servidores que processam grandes quantidades de **e-mails.**
- **Quick Spam Filter**: Filtro Bayesiano pequeno e rápido.
- **SpamAssassin**: Filtro que utiliza uma vasta gama de testes de heurísticas para determinar quando uma mensagem é **spam**. As mensagens que receberem uma "**nota**" menor do que um certo índice são classificados como **spam**.
- Comparado aos outros filtros, o filtro **SpamAssassin** é computacionalmente muito pesado, consome muitos recursos de máquinas.

Recomendações para evitar SPAM: http://www.dcc.unicamp.br/~jeronimo/spam.pt_BR.html
Destacar métodos para evitar SPAM
Filtros, listas de bloqueio e configurações com Postfix.
O artigo (http://www.linorg.cirp.usp.br/SSI/SSI2004/Artigos/Art017_ssi04.pdf) dá uma boa descrição do **greylist** e de outras técnicas para evitar SPAM.
Mais referências sobre SPAM:
http://www.benzedrine.cx/relaydb.html
http://spamlinks.net/filter-bl.htm

3.4 Greylist

- É uma modalidade de filtro que não utiliza nenhuma heurística ou outro método de análise estatística.
- Esse método de bloqueio de **spam** se baseia no comportamento do servidor que está enviando a mensagem ao invés do conteúdo da mensagem. O nome **"lista grey"** vem de uma mistura entre lista branca e preta com manutenção automática.
- O Software Postgrey é dedicado a eliminar **SPAMS** através da criação e manutenção de listas de Servidores SMTP armazenados em listas **WHITE, GREY e BLACK**.

- Seu funcionamento enquadra-se no meio termo entre o **whitelisting** (endereços de onde se aceita incondicionalmente as mensagens) e **blacklisting** (endereços de onde devem ser rejeitadas as mensagens). O que é feito é uma postergação da entrega da mensagem, usando o próprio protocolo SMTP.
- Quando uma caixa postal recebe uma nova mensagem de um contato desconhecido (IP não listado nas tabelas do sistema), a mensagem é temporariamente rejeitada com uma mensagem **"tente novamente mais tarde"** (isto ocorre na camada STMP e é transparente para o usuário final).
- É armazenado em um **banco de dados** um conjunto de informações suficientes para identificar unicamente cada mensagem.
- Em um curto intervalo de tempo, servidores de STMP, que aderem corretamente aos padrões do protocolo, farão uma nova tentativa de envio. Ao reenviar, o sistema encontrará as informações da mensagem na base de dados, liberando sua entrega.
- O protocolo SMTP é considerado como protocolo de transporte não-confiável, portanto, a possibilidade de falhas temporárias estão embutidas no núcleo do protocolo (RFC 821).
- Todo agente de transferência de mensagens (MTA) bem implementado promove tentativas de entrega de uma mensagem que tenha obtido um código de falha temporária. Isso geralmente ocorre, por exemplo, quando a fila de um servidor destino está muito longa para ser processada, ou o servidor tem uma carga muito alta (de operações de I/O). É neste aspecto que o **greylisting** é bem sucedido.
- Para ser rentável, um **spammer** utiliza um MTA desenvolvido para desconsiderar mensagens de erro no destino. Por exemplo, servidores recebem uma carga de mensagens que devem ser entregues a usuários gerados a partir de uma lista de nomes. Processar cada mensagem com falha geraria custo para o **spammer**, consumiria largura de banda no sentido oposto (de volta para ele).
- É mais rentável desprezar quaisquer mensagens de erro. Além disso, trocam constantemente de IP de origem, buscando burlar as **blacklists**. A não ser que estejam usando um MTA compatível com a especificação, não deverão fazer uma nova tentativa de entrega.
- O funcionamento do **greylist** é bastante simples e baseia-se em triplas contendo as seguintes informações:

1. endereço IP do servidor de **e-mail** que está enviando a mensagem
2. endereço do emissor de **e-mail**
3. endereço do receptor de **e-mail**

- Depois que o servidor de **e-mail** tenta enviar a mensagem é atribuído um tempo (em geral 30 minutos). Após esse tempo as mensagens são aceitas pelo servidor de destino. Então é gerada uma nova tripla contendo o endereço do servidor que enviou a mensagem e um tempo de 36 dias. Durante esses 36 dias o servidor poderá enviar mensagens sem ser barrado. Esse tempo é atribuído para que não haja um estouro da base de dados que armazena essa triplas. No caso de envio de uma nova mensagem esse tempo de 36 dias é renovado.

Referências sobre o **greylisting**:
http://www.comp.ufla.br/laboratorios/greylist.php

3.4.1 Instalação do Postgrey

No Linux Debian a instalação é bem simples:
$> apt-get install postgrey

3.4.2 Configurações e utilização com o MTA Postfix

Deve-se editar o arquivo de configuração do Postfix, **/etc/postfix/main.cf**, e acrescentar o item destacado:

smtpd_recipient_restrictions =
permit_mynetworks,
permit_sasl_authenticated,
reject_unknown_recipient_domain,
reject_unauth_pipelining,
reject_unauth_destination,
check_policy_service inet:127.0.0.1:60000,
check_policy_service unix:private/policy-spf

A porta de conexão é 60000 e apenas para o host local (IP 127.0.0.1)

No script **/etc/init.d/postgrey** há mais detalhes do controle de inicialização do **postgrey**.

Pode-se também criar listas que vão liberar **(whitelist)** automaticamente **e-mails** de certos clientes e domínios.

- Os arquivos estão no diretório **/etc/postgrey:**

whitelist_recipients:

postgrey whitelist for mail recipients
put this file in /etc/postgrey or specify its path
with –whitelist-recipients=xxx no arquivo **/etc/init.d/postgrey**
postmaster@
abuse@

whitelist_clients :

postgrey whitelist for mail client hostnames
——————————
put this file in /etc/postgrey or specify its path
with –whitelist-clients=xxx no arquivo **/etc/init.d/postgrey**
Debian-specific additions
gluck.debian.org
haydn.debian.org
klecker.debian.org
master.debian.org
merkel.debian.org
murphy.debian.org
newmurphy.debian.org
newraff.debian.org
spohr.debian.org
greylisting.org: Southwest Airlines (unique sender, no retry)
southwest.com
greylisting.org: Yahoo Groups servers (no retry)
scd.yahoo.com
greylisting.org: isp.belgacom.be (wierd retry pattern)
isp.belgacom.be
greylisting.org: Ameritrade (no retry)
ameritradeinfo.com
greylisting.org: Amazon.com (unique sender with letters)
amazon.com
2004-05-20: Linux kernel mailing-list (unique sender with letters)
vger.kernel.org
2004-06-02: karger.ch, no retry
karger.ch
2004-06-02: lilys.ch, (slow: 4 hours)

```
server-x001.hostpoint.ch
# 2004-06-09: roche.com (no retry)
gw.bas.roche.com
# 2004-06-09: newsletter (no retry)
mail.hhlaw.com
# 2004-06-09: no retry (reported by Ralph Hildebrandt)
prd051.appliedbiosystems.com
# 2004-06-17: swissre.com (no retry)
swissre.com
# 2004-06-17: dowjones.com newsletter (unique sender with letters)
returns.dowjones.com
# 2004-06-18: switch.ch (works but personnel is confused by the error)
domin.switch.ch
# 2004-06-23: accor-hotels.com (slow: 6 hours)
accor-hotels.com
# 2004-06-29: rr.com (no retry, reported by Duncan Hill)
/^ms-smtp.*\.rr\.com$/
# 2004-06-29: cox.net (no retry, reported by Duncan Hill)
/^lake.*mta.*\.cox\.net$/
# 2004-06-29: motorola.com (no retry)
mot.com
# 2004-07-01: nic.fr (address verification, reported by Arnaud Launay)
nic.fr
# 2004-07-01: verizon.net (address verification, reported by Bill Moran and Eric)
/^sc\d+pub\.verizon\.net$/
# 2004-07-02: cs.columbia.edu (no retry)
cs.columbia.edu
# 2004-07-02: papersinvited.com (no retry)
66.216.126.174
# 2004-07-02: telekom.de (slow: 6 hours)
/^mail\d+\.telekom\.de$/
# 2004-07-04: tiscali.dk (slow: 12 hours, reported by Klaus Alexander Seistrup)
/^smtp\d+\.tiscali\.dk$/
# 2004-07-04: freshmeat.net (address verification)
freshmeat.net
# 2004-07-11: zd-swx.com (unique sender with letters, reported by Bill Landry)
zd-swx.com
# 2004-07-11: lockergnome.wc09.net (unique sender with letters, reported by Bill Landry)
lockergnome.wc09.net
# 2004-07-19: mxlogic.net (no retry, reported by Eric)
p01m168.mxlogic.net
p02m169.mxlogic.net
# 2004-09-08: intel.com (pool on different subnets)
/^fmr\d+\.intel\.com$/
# 2004-09-17: cox-internet.com (no retry, reported by Rod Roark)
/^fe\d+\.cox-internet\.com$/
# 2004-10-11: logismata.ch (no retry)
logismata.ch
# 2004-11-25: brief.cw.reum.de (no retry, reported by Manuel Oetiker)
brief.cw.reum.de
# 2004-12-03: ingeno.ch (no retry)
qmail.ingeno.ch
# 2004-12-06: rein.ch (no retry)
mail1.thurweb.ch
# 2005-01-26: tu-ilmenau.de (no retry)
piggy.rz.tu-ilmenau.de
```

O seguinte artigo mostra as configurações de filtros para controle de Spam e Vírus. São utilizados Filtros Bayesianos, Sistema Antivírus Clamav, Controle com Greylist e Servidor MTA Postfix.

Os resultados são analisados de acordo com os seguintes itens:
Filtro aplicado
Número de mensagens recebidas
Número de mensagens com Falso Positivo
Número de mensagens com Falso Negativo
Índice de acerto
Mensagens recebidas
Mensagens entregues
Mensagens bloqueadas

http://www.linorg.cirp.usp.br/SSI/SSI2004/Artigos/Art017_ssi04.pdf

3.5 Servidores de E-Mails com o SpamAssassin, Postgrey e Postfix

Exemplos de instalação:

QUARTZO.CIRP.USP.BR Sistema Operacional : FreeBSD
SMTP - Sendmail
Anti-SPAM : SpamAssassin
Antivírus: McAfee
Resultados de Bloqueios de SPAM e Vírus **: /var/virusmails/**
WEB: http://143.107.200.45/Virus/
Aplicativo para relatório de tráfego de E-Mails : **ISOQLOG**

GLETE.EERP.USP.BR Sistema Operacional : FreeBSD 6.2
SMTP - Postfix
Anti-SPAM: SpamAssassin e Greylist POSTGREY
Antivírus : Clamav
Resultados de Bloqueios de SPAM e Vírus **: /var/virusmails/**
Aplicativo para relatório de tráfego de E-Mails : **PFLOGSUMM**

Instalação e configuração do SpamAssassin Recomendações, regras, etc:
http://spamassassin.apache.org

Greylist com SPAMD implementado no CIRP e FFCLRP:

1. http://granito2.cirp.usp.br/greylist/
2. http://143.107.138.130/spamd/

Anti-SPAM com POSTGREY:

1. http://www.security.usp.br/palestras/Greylisting_esecom.pdf
2. http://www.antispam.br/admin/greylisting/

Exemplos práticos de configuração de servidor de E-Mails com Postfix, Antivírus e Antispam:

1. http://granito2.cirp.usp.br/Antispam.Configs.html
2. http://semente.taurinus.org/fazenda/ConfigurandoPostfix

3.6 SPF - Sender Policy Framework

SPF é uma tecnologia para combater a falsificação de endereços de retorno dos emails (return-path).

O mecanismo permite:

- ao administrador de um domínio: definir e publicar uma política SPF, onde são designados os endereços das máquinas autorizadas a enviar mensagens em nome deste domínio; e
- ao administrador de um serviço de e-mail: estabelecer critérios de aceitação de mensagens em função da checagem das políticas SPF publicadas para cada domínio.

O processo de publicação de uma política SPF é independente da implantação de checagem de SPF por parte do MTA, estes podem ou não ser feitos em conjunto.

Mais detalhes: http://www.antispam.br/admin/spf

Capítulo 4

POP, IMAP e WEBMAIL

Todo servidor de **E-Mails** deve disponibilizar serviços para que os clientes possam usar seus **E-Mails**.

Os protocolos **POP3 e IMAP** são responsáveis pelo transporte de mensagens recebidas do servidor de **e-mail** para o cliente de **e-mail** do usuário. O protocolo **POP3** é mais antigo e mais simples, mas é mais popular e praticamente todos os programas clientes de **e-mail** o suportam. O protocolo **IMAP** é mais novo e possui mais funções que o **POP3**, no entanto nem todos os programas clientes de **e-mail** o suportam.

O suporte a estes protocolos não é feito pelo **Postfix**, mas sim por outros servidores.

4.0.1 Instalação

Para implementar um servidor **POP/IMAP** é necessário somente que um servidor de **e-mail** esteja instalado e configurado.

Será necessário também que o serviço **inetd** (ou **xinetd**) esteja ativo.

Instale os pacotes **uw-imapd e qpopper:**

$> apt-get install uw-imapd qpopper

Podem ser instalados outros pacotes de **POP e IMAP. Os mais comuns são:**

POP:

$> apt-cache seach pop

ipopd - POP2 and POP3 servers from UW
ipopd-ssl - POP2 and POP3 servers from UW
qpopper - Enhanced Post Office Protocol server (POP3).
qpopper-drac - Qpopper with DRAC Support
popa3d - A tiny POP3 daemon, designed with security as the primary goal
cyrus-pop3d - CMU Cyrus mail system (POP3 support)
courier-pop - POP3 daemon with PAM and Maildir support
courier-pop-ssl - POP3 daemon with SSL, PAM and Maildir support

IMAP:

$> apt-cache search imapd

cyrus-common - CMU Cyrus mail system (common files)
cyrus-imapd - CMU Cyrus mail system (IMAP support)
teapop - Powerful and flexible RFC-compliant POP3 server
teapop-mysql - Powerful and flexible RFC-compliant POP3 server
teapop-pgsql - Powerful and flexible RFC-compliant POP3 server
uw-imapd - remote mail folder access server
uw-imapd-ssl - remote mail folder access server

Para configurar uma conta **POP** não é necessária mais nenhuma configuração: ela usa as contas de usuários do sistema, ou as contas criadas pelo administrador **IMAP**.

4.1 Testando a Configuração

Para testar a configuração do **IMAP**, será necessário configurar um cliente de **e-mail** localizado em alguma máquina de sua rede para buscar suas mensagens no servidor. Mande algumas mensagens para um usuário e depois tente recuperar as mensagens deste usuário através do cliente de **e-mail.** Caso não seja possível recuperar essas mensagens verifique os arquivos de registro **/var/log/messages** e **/var/log/maillog** em busca de mensagens de erro.

O teste de funcionamento da configuração do **POP** pode ser feito da mesma forma: configure um cliente de **e-mail** para que ele busque as mensagens de uma conta **POP**; envie uma mensagem para esta conta e depois tente recuperá-la. Caso não funcione, verifique se a senha foi digitada corretamente e se a configuração do cliente de **e-mail** está correta.

Outro teste da configuração do **POP** também é simples, e pode ser feito no próprio servidor:

$> telnet localhost 110

Como o IMAP trabalha com a porta 143, faça o seguinte:

$> telnet localhost 143

Se não obter as mensagens:

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Os serviços não estão funcionando corretamente e as configurações devem ser verificadas.

Consulte o arquivo /etc/services para ver em quais portas estes serviços devem trabalhar:

$> grep imap /etc/services
$> grep pop /etc/services

Use o programa **nmap** também para fazer os testes :

$> nmap localhost

Starting nmap V. 2.54BETA31 (www.insecure.org/nmap/)
Interesting ports on localhost (127.0.0.1):
(The 1538 ports scanned but not shown below are in state: closed)
Port State Service
9/tcp open discard
13/tcp open daytime
21/tcp open ftp
22/tcp open ssh
25/tcp open smtp
37/tcp open time
53/tcp open domain
80/tcp open http
110/tcp open pop-3
111/tcp open sunrpc
113/tcp open auth
143/tcp open imap2

Pode-se usar também os comandos **netstat** ou **lsof** para verificar as portas **TCP/UDP** que estão sendo usadas no momento, da seguinte maneira:

$> lsof -n -i -P
$> netstat -natp

4.2 Configuração do Webmail

A instalação do **Webmail** é bastante simples. O **Squirrelmail** já faz parte dos pacotes das distribuições Debian. Pode-se instalar outro **Webmail**, tal como o **Openwebmail** ou o **IMP Horde**.

O **Squirrelmail** é baseado em linguagem **PHP**. Instale os dois pacotes da seguinte forma:

$> apt-get install php5 squirrelmail

Configure o **Squirremail** da seguinte maneira:

$> /etc/squirrelmail.conf.pl

Estabeleça os nomes do **Domínio**, do **SMTP**, do **IMAP** e suas respectivas portas de conexão.

Como descrito anteriormente, foi instalado o **IMAP uw-imapd.**

Configure a linguagem padrão (**pt_BR**), as figuras que devem aparecer na tela inicial (logotipo), nome do servidor, etc.

Personalize as configurações e, consulte sempre o site do **Squirrelmail** para conhecer outros recursos que podem ser implementados, tais como troca de senhas, domínios virtuais, utilização com Banco de dados, etc.

Instale o servidor **WEB Apache:**

$> apt-get install apache

Se preferir o **Openwebmail**, que depende do **Perl** e não está nos pacotes distribuídos pelo Debian Linux, versão **stable**, será necessário obter os fontes a partir do site (**http://www.openwebmail.org**). Este pacote está disponível para a distribuição **testing** do Debian Linux:

http://packages.debian.org/testing/web/openwebmail

* Depois de instalado o **Squirrelmail**, usando um **Browser** acesse o seguinte URL:

http://servidor.com.br/squirrelmail

* Se necessário, confira as configurações de seu servidor **Apache**. Verifique se os módulos necessários para o **PHP4** funcionar estão presentes no arquivo **httpd.conf**.

$> grep php /etc/apache/conf/httpd.conf

A resposta deve ser linhas com algo do tipo:

LoadModule php5_module /usr/lib/apache/1.3/libphp5.so
DirectoryIndex login.php index.html index.php index.htm index.shtml index.cgi
#AddType application/x-httpd-php5 .php5
#AddType application/x-httpd-php5-source .phps
#AddType application/x-httpd-php .php
#AddType application/x-httpd-php-source .phps

* Se não houver estas linhas, elas terão que ser adicionadas no **httpd.conf** e o servidor **Apache** deve ser reinicalizado.
* Observe os **PATHS** dos arquivos que tem que ser carregados por **LoadModule.**

Faca um teste para ver se o PHP5 está funcionando. Crie um script em PHP (teste.php) com o seguinte conteúdo:

<?
phpinfo();
?>

Salve-o no **/var/www/**

Aponte o **Browser** para o **script** da seguinte maneira:

http://servidor.com.br/teste.php

O resultado deve ser uma página bem extensa mostrando todas as configurações do **PHP** instalado.

Caso seja mostrado **apenas o conteúdo** do **script teste.php**, ou se for pedido para especificar o lugar onde salvar o arquivo, verifique as configurações do **PHP5** e do servidor **Apache**. Alguma configuração no **Apache** não está correta.

A cada modificação nas configurações do servidor **Apache,** lembre-se de reiniciá-lo com o comando:
$> /etc/init.d/apache restart

*** Caso não consiga ter resposta do servidor verifique se o Postfix, inetd (ou xinetd), popd, imapd e Apache estão ativos.**

Lembre-se de habilitar os serviços **popd e imapd** no **TCP WRAPPER**, nos arquivos **/etc/hosts.allow e /etc/hosts.deny.**

Veja os detalhes de como isso deve ser feito, verifique as sintaxes necessárias, etc.

Para acompanhar a evolução das mensagens de **log**, verifique constantemente os arquivos de **log**, mais especificamente os arquivos **/var/log/auth.log, /var/log/messages, /var/log/syslog, /var/log/apache/access.log e error.log, /var/log/mail/.**

Use os comandos :

$> tail -f /var/log/mail/mail.log
$> tail -f /var/log/mail/mail.err
$> tail -f /var/log/mail/mail.info
$> tail -f /var/log/mail/mail.warn

Faça o mesmo para os outros arquivos de **log.** Veja os **logs** do servidor **Apache** e consulte as mensagens de erro. Elas servem para lhe orientar melhor na busca de problemas de configuração.