

## Servidor de Correio Eletrônico com Sendmail, Cyrus IMAP, SpamAssassin e ClamAV no Fedora Core 6

### Índice

Servidor de Correio Eletrônico com Sendmail, Cyrus IMAP, SpamAssassin e ClamAV no Fedora Core 6.....	1
Termos de Uso.....	1
Introdução.....	2
Instalação.....	2
Configuração.....	2
Conexão Segura com TLS/SSL.....	14
Exemplo: Criação de Vários Usuários.....	16
Integrando o SpamAssassin.....	20
Instalação.....	20
Configuração.....	20
Configurando o Razor.....	22
Integrando o ClamAV.....	24
Instalação.....	24
Configuração.....	24
Monitorando as Mensagens com o Mailgraph.....	26
Referências.....	27

### Termos de Uso

#### Nota de Copyright

```
Copyright (c) 2007 Linux2Business.  
Permission is granted to copy, distribute and/or modify this document under  
the terms of the GNU Free Documentation License, Version 1.2 or any later  
version published by the Free Software Foundation; with no Invariant  
Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the  
license is included in the section entitled "GNU Free Documentation  
License".
```

O conteúdo textual deste documento está disponível sob a licença GNU FDL. Para maiores informações acesse <http://www.gnu.org/copyleft/fdl.html>. Os logotipos, marcas registradas e símbolos usados neste livro são de propriedade de seus respectivos proprietários.

## Introdução

Neste artigo veremos como instalar o **Sendmail** e o **Cyrus IMAP Server** no Fedora Core 6, com recursos como: Autenticação SMTP, Conexão SMTP Criptografada/Segura e Conexão POP3/IMAP Criptografada/Segura.

Vamos integrar também o **SpamAssassin** e o **ClamAV**, além de utilizar o **MailGraph** para monitoramento das mensagens recebidas e enviadas pelo nosso servidor.

## Instalação

Para instalar, utilizaremos o comando `yum`, como mostrado abaixo:

```
# yum install sendmail sendmail-cf
# yum install cyrus-imapd cyrus-imapd-utils cyrus-sasl-plain
```

Algumas dependências que provavelmente serão instaladas são: `m4` e `perl-Cyrus`.

**Obs.:** Não é necessário dizer que devemos ter acesso a internet para usar o comando `yum`, senão é preciso instalar os pacotes a partir dos CDs do Fedora Core 6.

## Configuração

O arquivo de configuração do **Sendmail** é `sendmail.mc` e fica localizado no diretório `/etc/mail`. Antes de começar a configuração é sempre interessante manter uma cópia de segurança deste arquivo:

```
# cd /etc/mail
# cp -p sendmail.mc sendmail.mc.default
```

A partir do arquivo original, vamos inserir alguns parâmetros e alterar outros, conforme apresentado abaixo:

```
divert(-1)dnl
dnl #
dnl # This is the sendmail macro config file for m4. If you make changes to
dnl # /etc/mail/sendmail.mc, you will need to regenerate the
dnl # /etc/mail/sendmail.cf file by confirming that the sendmail-cf package
dnl # is installed and then performing a
dnl #
dnl #     make -C /etc/mail
dnl #
include(`/usr/share/sendmail-cf/m4/cf.m4')dnl
VERSIONID(`setup for linux')dnl
OSTYPE(`linux')dnl
dnl #
dnl # SmtP Greeting Message
dnl #
define(`confSMTP_LOGIN_MSG', `$_j')dnl
dnl #
dnl # Double Bounce will be dropped
```

```

dnl #
define(`confDOUBLE_BOUNCE_ADDRESS', `bounce')dnl
dnl #
dnl # default logging level is 9, you might want to set it higher to
dnl # debug the configuration
dnl #
dnl define(`confLOG_LEVEL', `9')dnl
dnl #
dnl # Uncomment and edit the following line if your outgoing mail needs to
dnl # be sent out through an external mail server:
dnl #
dnl define(`SMART_HOST', `smtp.your.provider')dnl
dnl #
define(`confDEF_USER_ID', ``8:12'')dnl
dnl define(`confAUTO_REBUILD')dnl
define(`confTO_CONNECT', `lm')dnl
define(`confTRY_NULL_MX_LIST', true)dnl
define(`confDONT_PROBE_INTERFACES', true)dnl
define(`confPROCMAIL_MAILER_PATH', `/usr/bin/procmail')dnl
define(`ALIAS_FILE', `/etc/aliases')dnl
define(`STATUS_FILE', `/var/log/mail/statistics')dnl
define(`UUCP_MAILER_MAX', `2000000')dnl
define(`confUSERDB_SPEC', `/etc/mail/userdb.db')dnl
define(`confPRIVACY_FLAGS', `authwarnings, novrfy, noexpn, restrictgrun')dnl
define(`confAUTH_OPTIONS', `A')dnl
dnl #
dnl # The following allows relaying if the user authenticates, and
dnl # disallows plaintext authentication (PLAIN/LOGIN) on non-TLS links
dnl #
dnl define(`confAUTH_OPTIONS', `A p')dnl
dnl #
dnl # PLAIN is the preferred plaintext authentication method and used by
dnl # Mozilla Mail and Evolution, though Outlook Express and other MUAs do
dnl # use LOGIN. Other mechanisms should be used if the connection is not
dnl # guaranteed secure.
dnl # Please remember that saslauthd needs to be running for AUTH.
dnl #
TRUST_AUTH_MECH(`DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
dnl #
dnl # Rudimentary information on creating certificates for sendmail TLS:
dnl #     cd /usr/share/ssl/certs; make sendmail.pem
dnl # Complete usage:
dnl #     make -C /usr/share/ssl/certs usage
dnl #
dnl define(`confCACERT_PATH', `/etc/pki/tls/certs')dnl
dnl define(`confCACERT', `/etc/pki/tls/certs/ca-bundle.crt')dnl
dnl define(`confSERVER_CERT', `/etc/pki/tls/certs/sendmail.pem')dnl
dnl define(`confSERVER_KEY', `/etc/pki/tls/certs/sendmail.pem')dnl
dnl #
dnl # This allows sendmail to use a keyfile that is shared with OpenLDAP's
dnl # slapd, which requires the file to be readable by group ldap
dnl #

```

```

dnl define(`confDONT_BLAAME_SENDMAIL',`groupreadablekeyfile')dnl
dnl #
dnl define(`confTO_QUEUEWARN',`4h')dnl
dnl define(`confTO_QUEUERETURN',`5d')dnl
dnl define(`confQUEUE_LA',`12')dnl
dnl define(`confREFUSE_LA',`18')dnl
define(`confTO_IDENT',`0')dnl
FEATURE(delay_checks)dnl
FEATURE(`no_default_msa',`dnl')dnl
FEATURE(`smrsh',`/usr/sbin/smrsh')dnl
FEATURE(`mailertable',`hash -o /etc/mail/mailertable.db')dnl
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable.db')dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
dnl #
dnl # The following limits the number of processes sendmail can fork to
dnl # accept incoming messages or process its message queues to 12.)
dnl # sendmail refuses to accept connections once it has reached its quota
dnl # of child processes.
dnl #
define(`confMAX_DAEMON_CHILDREN',`12)dnl
dnl #
dnl # Limits the number of new connections per second. This caps the
dnl # overhead
dnl # incurred due to forking new sendmail processes. May be useful against
dnl # DoS attacks or barrages of spam. (As mentioned below, a per-IP
dnl # address
dnl # limit would be useful but is not available as an option at this
dnl # writing.)
dnl #
define(`confCONNECTION_RATE_THROTTLE',`3)dnl
dnl #
dnl # The -t option will retry delivery if e.g. the user runs over his
dnl #
FEATURE(local_procmail,`,`procmail -t -Y -a $h -d $u')dnl
FEATURE(`access_db',`hash -T<TMPF> -o /etc/mail/access.db')dnl
FEATURE(`blacklist_recipients')dnl
EXPOSED_USER(`root')dnl
dnl #
dnl # For using Cyrus-IMAPd as POP3/IMAP server through LMTP delivery
dnl # uncomment the following 2 definitions and activate below in the
dnl # MAILER section the cyrusv2 mailer.
dnl #
define(`confLOCAL_MAILER',`cyrusv2')dnl
define(`CYRUSV2_MAILER_ARGS',`FILE /var/lib/imap/socket/lmtp')dnl
dnl #
dnl # The following causes sendmail to only listen on the IPv4 loopback
dnl # address and not on any other network devices. Remove the loopback
dnl # address restriction to accept email from the internet or intranet.
dnl #
DAEMON_OPTIONS(`Port=smtp, Name=MTA')dnl

```

```
dnl #
dnl # The following causes sendmail to additionally listen to port 587 for
dnl # mail from MUAs that authenticate. Roaming users who can't reach their
dnl # preferred sendmail daemon due to port 25 being blocked or redirected
dnl # find
dnl # this useful.
dnl #
dnl DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 465, but
dnl # starting immediately in TLS mode upon connecting. Port 25 or 587
dnl # by STARTTLS is preferred, but roaming clients using Outlook Express
dnl # can't do STARTTLS on ports other than 25. Mozilla Mail can ONLY use
dnl # STARTTLS and doesn't support the deprecated smtps; Evolution
dnl # <1.1.1 uses smtps when SSL is enabled-- STARTTLS support is
dnl # available in version 1.1.1.
dnl #
dnl # For this to work your OpenSSL certificates must be configured.
dnl #
dnl DAEMON_OPTIONS(`Port=smtps, Name=TLSMTA, M=s')dnl
dnl #
dnl # The following causes sendmail to additionally listen on the Ipv6
dnl # loopback device. Remove the loopback address restriction listen to
dnl # the network.
dnl #
dnl DAEMON_OPTIONS(`port=smtp,Addr>:::1, Name=MTA-v6, Family=inet6')dnl
dnl #
dnl # enable both ipv6 and ipv4 in sendmail:
dnl #
dnl DAEMON_OPTIONS(`Name=MTA-v4, Family=inet, Name=MTA-v6, Family=inet6')
dnl #
dnl # We strongly recommend not accepting unresolvable domains if you want
dnl # to protect yourself from spam. However, the laptop and users on
dnl # computers that do not have 24x7 DNS do need this.
dnl #
dnl FEATURE(`accept_unresolvable_domains')dnl
dnl #
dnl FEATURE(`relay_based_on_MX')dnl
dnl #
dnl # Also accept email sent to "localhost.localdomain" as local email.
dnl #
LOCAL_DOMAIN(`localhost.localdomain')dnl
dnl #
dnl # The following example makes mail from this host and any additional
dnl # specified domains appear to be sent from mydomain.com
dnl #
MASQUERADE_AS(`linux2business.net.br')dnl
dnl #
dnl # masquerade not just the headers, but the envelope as well
dnl #
FEATURE(masquerade_envelope)dnl
dnl #
dnl # masquerade not just @mydomainalias.com, but @*.mydomainalias.com as
```

```
well
dnl #
dnl FEATURE(masquerade_entire_domain)dnl
dnl #
dnl MASQUERADE_DOMAIN(localhost)dnl
dnl MASQUERADE_DOMAIN(localhost.localdomain)dnl
dnl MASQUERADE_DOMAIN(mydomainalias.com)dnl
dnl MASQUERADE_DOMAIN(mydomain.lan)dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
MAILER(cyrusv2)dnl
```

Para que o **Sendmail** entregue as mensagens localmente é preciso adicionar no arquivo `local-host-names`, localizado no diretório `/etc/mail`, o domínio do servidor, no nosso caso, `linux2business.net.br`. Sem isto, as mensagens não serão entregues e nos registros das atividades, no arquivo `/var/log/maillog`, será apresentados erros relacionados ao DNS e o campo MX.

```
# cd /etc/mail
# echo "linux2business.net.br" >> local-host-names
```

Para testar, podemos usar o comando `sendmail`:

```
# sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> $=w
linux2business.net.br
fc5
fc5.linux2business.net.br
localhost.localdomain
localhost
[127.0.0.1]
> /quit
```

**Obs.:** É preciso utilizar o parâmetro `use_cw_file`, no arquivo de configuração do Sendmail, para que o arquivo `local-host-names` tenha efeito.

Por último, para aumentar a segurança, vamos criar o arquivo `virtusertable`, localizado no diretório `/etc/mail`, que permite múltiplos domínios virtuais em um mesmo servidor. No nosso caso, onde não temos domínios virtuais, serve para controlar os endereços de destino das mensagens.

Neste arquivo iremos adicionar todos os usuários e *alias* que desejamos entregar mensagens. Conforme apresentado abaixo:

```
# cd /etc/mail
# cat > virtusertable
user1@linux2business.net.br user1
user2@linux2business.net.br user2
alias1@linux2business.net.br alias1
@linux2business.net.br error:nouser User address is not valid
```

```
Ctrl-D
```

Depois devemos reiniciar o **Sendmail** ou executar o comando abaixo para criar o arquivo `virtusertable.db`, também localizado no diretório `/etc/mail`.

```
# cd /etc/mail
# makemap hash virtusertable.db < virtusertable
```

Para testar, podemos usar o comando `sendmail`:

```
# sendmail -bv user1@linux2business.net.br
user1@linux2business.net.br... deliverable: mailer cyrusv2, user user1
#
# sendmail -bv spam@linux2business.net.br
spam@linux2business.net.br... User address is not valid
```

Com a configuração realizada, precisamos agora gerar o arquivo `sendmail.cf` que é o arquivo de configuração utilizado de fato pelo **Sendmail**. Para isto, podemos utilizar várias maneiras, sendo algumas apresentadas abaixo:

- Através do comando `make`, que deve estar instalado:

```
# cd /etc/mail
# make
```

- Através do comando `m4`, que foi instalado como dependência do pacote `sendmail-cf`:

```
# cd /etc/mail
# m4 sendmail.mc > sendmail.cf
```

- Através do comando `service`, simplesmente iniciando ou reiniciando o **Sendmail**:

```
# service sendmail restart
```

**Obs.:** Qualquer alteração no arquivo `sendmail.mc` somente terá efeito quando o arquivo `sendmail.cf` for criado novamente.

Uma vez configurado o **Sendmail**, vamos iniciar o processo de configuração do **Cyrus IMAP**. O arquivo de configuração é `cyrus.conf` e fica localizado no diretório `/etc`.

```
# standard standalone server implementation

START {
  # do not delete this entry!
  recover          cmd="ctl_cyrusdb -r"

  # this is only necessary if using idled for IMAP IDLE
  idled cmd="idled"
}

# UNIX sockets start with a slash and are put into /var/lib/imap/sockets
SERVICES {
  # add or remove based on preferences
```

```
imap cmd="imapd" listen="imap" prefork=5
imaps cmd="imapd -s" listen="imaps" prefork=1
pop3 cmd="pop3d" listen="pop3" prefork=3
pop3s cmd="pop3d -s" listen="pop3s" prefork=1
sieve cmd="timsieved" listen="sieve" prefork=0

# these are only necessary if receiving/exporting usenet via NNTP
# nntp cmd="nntpd" listen="nntp" prefork=3
# nntpscmd="nntpd -s" listen="nntps" prefork=1

# at least one LMTP is required for delivery
# lmtp cmd="lmtpd" listen="lmtp" prefork=0
lmtpunix cmd="lmtpd" listen="/var/lib/imap/socket/lmtp" prefork=1

# this is only necessary if using notifications
# notify cmd="notifyd" listen="/var/lib/imap/socket/notify"
proto="udp" prefork=1
}

EVENTS {
# this is required
checkpoint cmd="ctl_cyrusdb -c" period=30

# this is only necessary if using duplicate delivery suppression,
# Sieve or NNTP
delprune cmd="cyr_expire -E 3" at=0400

# this is only necessary if caching TLS sessions
tlsprune cmd="tls_prune" at=0400
}
```

A configuração padrão habilita os serviços IMAP, IMAPS, POP3, POP3S e SIEVE, onde caso não seja necessário algum, basta comentar a linha referente, no campo SERVICES. A configuração padrão também habilita os serviços em todas as interfaces de comunicação e caso precise, basta acrescentar o endereço IP ao parâmetro `listen`, de acordo com a sintaxe ENDEREÇO:PORTA, para que somente uma determinada interface seja habilitada.

Uma configuração muito útil, seria o uso do comando `squatter`, localizado no diretório `/usr/lib/cyrus-imapd`, responsável por indexar as caixas postais, o que aumenta a performance do servidor. Para isto, acrescente a seguinte linha no campo EVENTS do arquivo de configuração:

```
squatter cmd="squatter -r user" period=1440
```

Para conhecer mais sobre as configurações possíveis, uma dica importante é o manual do arquivo de configuração, `man cyrus.conf`.

Em seguida devemos configurar o **SASL**, *Simple Authentication and Security Layer*, que irá prover um método de autenticação para os serviços **Sendmail** e **Cyrus IMAP**.

O arquivo de configuração é `imapd.conf`, localizado no diretório `/etc`. A partir do arquivo original, vamos inserir alguns parâmetros e alterar outros, conforme apresentado abaixo:

```
configdirectory: /var/lib/imap
partition-default: /var/spool/imap
admins: cyrus
sievedir: /var/lib/imap/sieve
sendmail: /usr/sbin/sendmail
hashimapspool: true
sasl_pwcheck_method: saslauthd
sasl_mech_list: PLAIN
tls_cert_file: /etc/pki/cyrus-imapd/cyrus-imapd.pem
tls_key_file: /etc/pki/cyrus-imapd/cyrus-imapd.pem
tls_ca_file: /etc/pki/tls/certs/ca-bundle.crt
allowplaintext: yes
autocreatequota: 102400
autocreateinboxfolders: Spam
```

Para fazer com que todas as mensagens marcadas como SPAM sejam enviadas automaticamente para a pasta Spam, podemos utilizar o parâmetro `autocreate_sieve_script`, passando como valor o caminho completo até um script Sieve, conforme abaixo:

```
require ["imapflags","fileinto"];
if header :comparator "i;ascii-casemap" :matches "X-Spam-Status" "Yes*" {
  addflag "\\Seen";
  fileinto "INBOX.Spam";
  removeflag "\\Seen";
}
```

O script acima foi copiado no diretório `/var/lib/imap/sieve/script` com o nome de `spam.sieve` e foi adicionado o seguinte parâmetro no arquivo `/etc/imapd.conf`:

```
autocreate_sieve_script: /var/lib/imap/sieve/script/spam.sieve
```

Outro arquivo importante para a configuração do **SASL** é `saslauthd`, localizado no diretório `/etc/sysconfig`. Neste arquivo é definido qual mecanismo será utilizado para autenticação das senhas. Por padrão, a autenticação é realizada por PAM e, por exemplo, para autenticação por LDAP, é preciso alterar o parâmetro `MECH` para `ldap`.

```
# Directory in which to place saslauthd's listening socket, pid file, and
# so on. This directory must already exist.
SOCKETDIR=/var/run/saslauthd

# Mechanism to use when checking passwords. Run "saslauthd -v" to get a
# list of which mechanism your installation was compiled with the ability to
# use.
MECH=pam

# Additional flags to pass to saslauthd on the command line. See
# saslauthd(8) for the list of accepted flags.
FLAGS=
```

Por último, devemos criar as caixas postais dos usuários no **Cyrus IMAP**, através do comando `cyradm`. Com ele é possível criar, remover, configurar permissões e quotas das caixas postais. Para

maiores informações, man cyradm.

**Obs.:** Para criar as caixas postais é preciso que os serviços saslauthd e cyrus-imapd estejam iniciados. É necessário também, que uma senha seja criada para o usuário cyrus.

- Criar Caixas Postais

```
# cyradm -u cyrus localhost
IMAP Password:
fc5.linux2business.net.br> cm user.guest
fc5.linux2business.net.br> lm user.guest
user.guest (\HasNoChildren)
```

- Remover Caixas Postais

```
# cyradm -u cyrus localhost
IMAP Password:
fc5.linux2business.net.br> sam user.guest cyrus x
fc5.linux2business.net.br> dm user.guest
```

- Configurar Quota

```
# cyradm -u cyrus localhost
IMAP Password:
fc5.linux2business.net.br> sq user.guest 102400
quota:102400
fc5.linux2business.net.br> lq user.guest
STORAGE 0/102400 (0%)
```

Após entrar no ambiente do cyradm, existe o comando help que mostra mais opções, conforme abaixo:

```
# cyradm -u cyrus localhost
IMAP Password:
fc5.linux2business.net.br> help
authenticate, login, auth authenticate to server
chdir, cd change current directory
createmailbox, create, cm create mailbox
deleteaclmailbox, deleteacl, dam remove ACLs from mailbox
deletemailbox, delete, dm delete mailbox
disconnect, disc disconnect from current server
exit, quitexit cyradm
help, ? show commands
info display mailbox/server metadata
listacl, lam, listaclmailbox list ACLs on mailbox
listmailbox, lm list mailboxes
listquota, lq list quotas on specified root
listquotaroot, lqr, lqm show quota roots and quotas for mailbox
mboxcfg, mboxconfig configure mailbox
reconstruct reconstruct mailbox (if supported)
renamemailbox, rename, renm rename (and optionally relocate) mailbox
server, servername, connect show current server or connect to server
setaclmailbox, sam, setaclset ACLs on mailbox
setinfo set server metadata
```

```
setquota, sq      set quota on mailbox or resource
version, ver     display version info of current server
xfermailbox, xfer transfer (relocate) a mailbox to a different server
```

Uma vez a configuração finalizada, vamos iniciar os serviços e realizar alguns testes, observando o que acontece através do arquivo de log /var/log/maillog.

```
# service sendmail start
# service cyrus-imapd start
# service saslauthd start
```

```
# netstat -nltp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 0.0.0.0:993 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:995 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:110 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:143 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:2000 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN 1517/sendmail
tcp 0 0 :::993 :::* LISTEN 1438/cyrus-master
tcp 0 0 :::995 :::* LISTEN 1438/cyrus-master
tcp 0 0 :::110 :::* LISTEN 1438/cyrus-master
tcp 0 0 :::143 :::* LISTEN 1438/cyrus-master
tcp 0 0 :::2000 :::* LISTEN 1438/cyrus-master
```

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 fc5.linux2business.net.br ESMTP
ehlo fc5
250-fc5.linux2business.net.br Hello fc5.linux2business.net.br [127.0.0.1],
pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
250-DELIVERBY
250 HELP
quit
Connection closed by foreign host.
```

```
# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

```

```
+OK fc5.linux2business.net.br Cyrus POP3 v2.3.1-Invoca-RPM-2.3.1-2.6.fc5
server ready <3388713543.1154044342@fc5.linux2business.net.br>
quit
+OK
Connection closed by foreign host.
```

```
# telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK fc5.linux2business.net.br Cyrus IMAP4 v2.3.1-Invoca-RPM-2.3.1-2.6.fc5
server ready
. logout
* BYE LOGOUT received
. OK Completed
Connection closed by foreign host.
```

```
# testsaslauthd -u user -p password
```

Para testar o envio e recebimento de mensagens, podemos utilizar o comando telnet, muito prático nos primeiros momentos de verificação do servidor.

- Enviando mensagem

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 fc5.linux2business.net.br ESMTP
ehlo fc5
250-fc5.linux2business.net.br Hello, pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
250-DELIVERBY
250 HELP
mail from:guest@linux2business.net.br
250 2.1.0 guest@linux2business.net.br... Sender ok
rcpt to:sandro@linux2business.net.br
250 2.1.5 sandro@linux2business.net.br... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Subject: Teste
Primeiro envio de mensagem para teste do ambiente.
.
250 2.0.0 k6S02qIV001814 Message accepted for delivery
quit
```

```
221 2.0.0 fc5.linux2business.net.br closing connection
Connection closed by foreign host.
```

- Recebendo mensagem por IMAP

```
# telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK fc5.linux2business.net.br Cyrus IMAP4 v2.3.1-Invoca-RPM-2.3.1-
2.6.fc5 server ready
01 login user password
01 OK User logged in
02 select inbox
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen *)]
* 1 EXISTS
* 1 RECENT
* OK [UNSEEN 1]
* OK [UIDVALIDITY 1154043337]
* OK [UIDNEXT 2]
02 OK [READ-WRITE] Completed
03 fetch 1 body[text]
* 1 FETCH (FLAGS (\Recent \Seen) BODY[TEXT] {52}
Primeiro envio de mensagem para teste do ambiente.
)
03 OK Completed (0.000 sec)
04 logout
* BYE LOGOUT received
04 OK Completed
Connection closed by foreign host.
```

- Recebendo mensagem por POP3

```
# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK fc5.linux2business.net.br Cyrus POP3 v2.3.1-Invoca-RPM-2.3.1-
2.6.fc5 server ready
<1273070995.1154046595@fc5.linux2business.net.br>
user user
+OK Name is a valid mailbox
pass password
+OK Mailbox locked and ready
list
+OK scan listing follows
1 663
.
retr 1
+OK Message follows
Return-Path: <guest@linux2business.net.br>
Received: from fc5.linux2business.net.br ([unix socket])
```

```
by fc5.linux2business.net.br (Cyrus v2.3.1-Invoca-RPM-2.3.1-2.6.fc5)
with LMTPA;
Thu, 27 Jul 2006 21:23:57 -0300
X-Sieve: CMU Sieve 2.3
Received: from fc5 (fc5.linux2business.net.br [127.0.0.1])
by fc5.linux2business.net.br (8.13.7/8.13.7) with ESMTTP id
k6S0N6EZ001878
for sandro@linux2business.net.br; Thu, 27 Jul 2006 21:23:26 -0300
Date: Thu, 27 Jul 2006 21:23:06 -0300
From: guest@linux2business.net.br
Message-Id: <200607280023.k6S0N6EZ001878@fc5.linux2business.net.br>
Subject: Teste2
```

Primeiro envio de mensagem para teste do ambiente.

```
.
quit
+OK
Connection closed by foreign host.
```

## Conexão Segura com TLS/SSL

O **SSL**, *Secure Socket Layer*, e, seu sucessor, **TLS**, *Transport Layer Security*, são protocolos de criptografia que provêem comunicação segura para correio eletrônico, internet e qualquer outro serviço que precise de transferência de arquivos.

A diferença básica entre **TLS** e **SSL**, está no fato de que ao utilizar o primeiro, nenhuma outra porta de comunicação é necessária, ou seja, tudo acontece na porta padrão, ou seja, 25/tcp para o **Sendmail**, 110/tcp e 143/tcp, respectivamente POP3 e IMAP, para o **Cyrus IMAP**. No **SSL** é preciso outra porta, normalmente a 465/tcp para o **Sendmail** e as portas 995/tcp e 993/tcp, respectivamente POP3S e IMAPS, para o **Cyrus IMAP**.

O primeiro passo seria a criação de um certificado, onde utilizaremos o pacote **OpenSSL**. Se este pacote não estiver instalado, é preciso instalá-lo, através do comando yum ou dos CDs do Fedora Core 6.

Para criação do certificado, tanto para o **Sendmail** quanto para **Cyrus IMAP**, vamos usar os seguintes comandos, que foram retirados do script `/etc/pki/tls/misc/CA`:

**Obs.:** Talvez alterações no arquivo `/etc/pki/tls/openssl.cnf` sejam necessárias, assim como a criação de alguns arquivos.

- Criação da Chave Privada, *Private Key*, e da Requisição de Certificado, *Certificate Request*:

```
# mkdir -p ./CA/certs
# mkdir ./CA/newcerts
# touch ./CA/index.txt
# echo "00" > ./CA/serial
# cd ./CA/certs
# openssl req -new -keyout key.pem -out req.pem
```

- Criação do Certificado de Autoridade, *Certificate Authority (CA)*, com a Chave Privada e a Requisição de Certificado:

```
# openssl ca -out cacert.pem -outdir . -days 3650 -batch -keyfile
key.pem -selfsign -infiles req.pem
```

- Criação da Requisição de Certificado para o servidor, *Certificate Signing Request (CSR)*:

```
# openssl req -newkey rsa:1024 -nodes -keyout serverkey.pem -out
serverkey.pem
```

- Criação do Certificado para o servidor, assinado pela CA, com a Chave Privada e a CSR:

```
# openssl ca -policy policy_anything -keyfile key.pem -cert
cacert.pem -out servercrt.pem -infiles serverkey.pem
```

**Obs.:** Alguns parâmetros utilizados para criação dos certificados e chaves, estão definidos no arquivo `/etc/pki/tls/openssl.cnf`.

Para finalizar, primeiro vamos definir as permissões e colocar os arquivos em um diretório apropriado tanto para o **Sendmail** quanto para o **Cyrus IMAP**:

```
# cd /etc/mail
# mkdir certs
# cd certs
# mv /path to file/cacert.pem .
# mv /path to file/servercrt.pem .
# mv /path to file/serverkey.pem .
# cp -p serverkey.pem sendmail.pem
# cp -p serverkey.pem cyrus-imapd.pem
# chown root.root cacert.pem servercrt.pem sendmail.pem
# chown cyrus.root cyrus-imapd.pem
# chmod 444 cacert.pem servercrt.pem
# chmod 400 sendmail.pem cyrus-imapd.pem
# rm serverkey.pem
```

Em seguida, vamos alterar o arquivo `sendmail.mc`, informando o caminho dos certificados e chaves criadas, além da porta `465/tcp`:

```
define(`confCACERT_PATH',`/etc/mail/certs')dnl
define(`confCACERT',`/etc/mail/certs/cacert.pem')dnl
define(`confSERVER_CERT',`/etc/mail/certs/servercrt.pem')dnl
define(`confSERVER_KEY',`/etc/mail/certs/sendmail.pem')dnl
DAEMON_OPTIONS(`Port=smtps, Name=TLSMTA, M=s')dnl
```

Um procedimento parecido deve ser realizado no arquivo `imapd.conf`:

```
tls_cert_file: /etc/mail/certs/servercrt.pem
tls_key_file: /etc/mail/certs/cyrus-imapd.pem
tls_ca_file: /etc/mail/certs/cacert.pem
```

Por último, vamos reiniciar os serviços:

```
# service sendmail restart
# service cyrus-imapd restart
```

Alguns testes podem ser realizados, observando sempre o que acontece através do arquivo de log /var/log/maillog.

```
# netstat -nltp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 0.0.0.0:993 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:995 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:110 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:143 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:2000 0.0.0.0:* LISTEN 1438/cyrus-master
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN 1517/sendmail:acc
tcp 0 0 0.0.0.0:465 0.0.0.0:* LISTEN 1517/sendmail:acc
tcp 0 0 :::993 :::* LISTEN 1438/cyrus-master
tcp 0 0 :::995 :::* LISTEN 1438/cyrus-master
tcp 0 0 :::110 :::* LISTEN 1438/cyrus-master
tcp 0 0 :::143 :::* LISTEN 1438/cyrus-master
tcp 0 0 :::2000 :::* LISTEN 1438/cyrus-master
```

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 fc5.linux2business.net.br ESMTP
ehlo fc5
250-fc5.linux2business.net.br Hello fc5.linux2business.net.br [127.0.0.1],
pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
Connection closed by foreign host.
```

### Exemplo: Criação de Vários Usuários

Caso você tenha uma lista com muitos usuários, fica inviável a criação manual destes, utilizando somente o comando cyradm. Para facilitar, existe disponível na internet um script, escrito em Perl, que realiza a criação automática de usuários, como apresentado abaixo:

(Nota: O script abaixo foi copiado da internet, porém infelizmente não tenho a referência para estar indicando aqui)

```
#!/usr/bin/perl -w
#
# This will create a new mailbox and set a quota on the new user. Just be
```

```
# sure that you installed the Cyrus::IMAP perl module.  If you did
# 'make all && make install' or installed Cyrus using the FreeBSD ports you
# don't have to do anything at all.
#
# Change the params below to match your mailserver settings, and
# your good to go!
#
# Author: amram@manhattanprojects.com
#
# modified by Tom Lazar tom@tomster.org on 2003-08-26 to use
# a tab separated user - passwd inputfile instead of the standardpassword

use Cyrus::IMAP::Admin;

#
# CONFIGURATION PARAMS
#
my $cyrus_server = "localhost";
my $cyrus_user = "cyrus";
my $cyrus_pass = "123456";

# 100 Megs
my $quota_size = "102400";

my $mechanism = "login";

#
# EOC
#

if (!$ARGV[0]) {
    die "Usage: $0 [user to add] \n";
    # die "Usage: $0 [user to add] passwd \n";
} else {
    $newuser = "$ARGV[0]";
    # $newpasswd = "$ARGV[1]";
}

sub createMailbox {

    my ($user, $subfolder) = @_ ;

    my $cyrus = Cyrus::IMAP::Admin->new($cyrus_server);
    $cyrus->authenticate($mechanism,'imap','',$cyrus_user,'0','10000',$cyrus_pass);

    if ($subfolder eq "INBOX") {
        $mailbox = "user.". $user;
    } else {
        $mailbox = "user.". $user .".". $subfolder;
    }

    $cyrus->create($mailbox);
}
```

```

if ($cyrus->error) {
    print STDERR "Error: ", $mailbox, " ", $cyrus->error, "\n";
} else {
    print "Created Mailbox: $mailbox \n";
}
}

sub setQuota {

    my ($user) = @_ ;

    my $cyrus = Cyrus::IMAP::Admin->new($cyrus_server);
    $cyrus->authenticate($mechanism, 'imap', '', $cyrus_user, '0', '10000', $cyrus_pass);

    $mailbox = "user.". $user;
    $cyrus->setquota($mailbox, "STORAGE", $quota_size);
    if ($cyrus->error) {
        print STDERR "Error: ", $mailbox, " ", $cyrus->error, "\n";
    } else {
        print "Setting Quota: $mailbox at $quota_size \n";
    }
}

print "Adding User: ", $newuser, "\n";

createMailbox($newuser, 'INBOX');
createMailbox($newuser, 'Spam');
# createMailbox($newuser, 'Trash');
# createMailbox($newuser, 'Drafts');
# createMailbox($newuser, 'Junk');

setQuota($newuser);

# This portion below will set a password for the user you wanted to
# add.

# system "echo ". $newpasswd ." > .saslpass.tmp";
# system "saslpasswd2 -p $newuser < .saslpass.tmp";
# print "Generated Password: Completed \n";
# unlink(".saslpass.tmp");

```

Como pode ser percebido, é possível realizar certas alterações no script acima, de acordo com suas necessidades. Um exemplo de um comando utilizando o script, está apresentado abaixo:

```

# for i in $(cat users.txt)
> do
> add_cyrus_user.pl $i
> sleep 1
> done

```

Uma outra maneira, mais fácil, de criar vários usuários, porém é necessário conhecer a senha dos usuários, o que as vezes é algo que também está sendo definido. Assim, um script, utilizando o comando abaixo tem o mesmo efeito que o script apresentado acima:

```
# echo ". logout" | imtest -a user -w password
```

As pastas que serão criadas e a quota do usuário devem estar previamente definidas no arquivo `imapd.conf`. Uma observação importante é sempre acompanhar tudo o que acontece, pelo arquivo `/var/log/maillog`.

## Integrando o SpamAssassin

### Instalação

Para instalar, utilizaremos o comando `yum`, como mostrado abaixo:

```
# yum install spamassassin spamass-milter
```

**Obs.:** Não é necessário dizer que devemos ter acesso a internet para usar o comando `yum`, senão é preciso instalar os pacotes a partir dos CDs do Fedora Core 6.

### Configuração

Para realizar a integração do **SpamAssassin** com o **Sendmail**, vamos acrescentar ao arquivo `/etc/mail/sendmail.mc` os seguintes parâmetros:

```
INPUT_MAIL_FILTER(`spamassassin', `S=unix:/var/run/spamass-milter/spamass-
milter.sock, F=, T=C:15m;S:4m;R:4m;E:10m')dnl
define(`confMILTER_MACROS_CONNECT', `t, b, j, _, {daemon_name}, {if_name},
{if_addr}')dnl
define(`confMILTER_MACROS_ENVRcpt', `r, v, Z')dnl
define(`confMILTER_MACROS_HELO', `s, {tls_version}, {cipher}, {cipher_bits},
{cert_subject}, {cert_issuer}')dnl
```

**Obs.:** As informações acima foram retiradas do arquivo *README*, localizado no diretório `/usr/share/doc/spamass-milter`.

Outro arquivo importante na configuração do **SpamAssassin**, na verdade do **SpamAssassin Milter**, está localizado no diretório `/etc/sysconfig` com o nome `spamass-milter`, onde neste arquivo podemos habilitar a variável `EXTRA_FLAGS`, simplesmente retirando o caractere `#` do início.

Os parâmetros contidos nesta variável configuram o **SpamAssassin** para não modificar o campo Assunto, entre outros, e também para rejeitar mensagens onde a pontuação ultrapasse 15 pontos:

```
EXTRA_FLAGS="-m -r 15"
```

**Obs.:** Para maiores detalhes sobre estes e outros parâmetros, seria interessante consultar o manual do **SpamAssassin Milter**:

```
# man spamass-milter
```

Por último, vamos configurar o **SpamAssassin**, alterando seu arquivo de configuração, `local.cf`, localizado no diretório `/etc/mail/spamassassin`, conforme apresentado abaixo:

```
# How many hits before a message is considered spam.
required_score          5

# Encapsulate spam in an attachment (0=no, 1=yes, 2=safe)
report_safe             0

# Enable the Bayes system
```

```

use_bayes          1

# Enable Bayes auto-learning
bayes_auto_learn  1

# Set up a systemwide Bayesian databases
bayes_path         /var/lib/spamassassin/bayes
bayes_file_mode   0644

# Set up a systemwide autowhitelist
auto_whitelist_path /var/lib/spamassassin/auto-whitelist
auto_whitelist_file_mode 0644

# Enable or disable network checks
skip_rbl_checks   0
use_razor2        0
use_pyzor         0

# Trusted Network (Alterar de acordo com os endereços utilizados)
trusted_networks  192.168.1.100
internal_networks 192.168.1.100

# Personal Rules
header  LOCAL_FROM_HOTMAIL  From =~ /hotmail\.com/i
score   LOCAL_FROM_HOTMAIL  1.0
describe LOCAL_FROM_HOTMAIL  From hotmail.com

header  LOCAL_CHARGES        Subject =~ /Charge para Voce/i
score   LOCAL_CHARGES        1.5
describe LOCAL_CHARGES        Fake charges.com.br

header  LOCAL_TE_AMO         Subject =~ /Te Amo/i
score   LOCAL_TE_AMO         1.5
describe LOCAL_TE_AMO         Virus

score   HTML_MESSAGE         1.0
score   BAYES_99              4.3
score   BAYES_95              3.5
score   BAYES_80              3.0

```

**Obs.:** Uma parte da configuração acima foi copiada da referência *SpamAssassin Configuration Generator*, que gera a configuração a partir das opções escolhidas. O restante foi feito, tomando como base a documentação do projeto *SpamAssassin*.

**Obs.:** Para um entendimento melhor sobre os parâmetros `trusted_networks` e `internal_networks` é recomendado a leitura do texto *TrustPath*, disponível no projeto do **SpamAssassin** e apresentado nas referência deste artigo.

É necessário criar o diretório `/var/lib/spamassassin`, conforme abaixo:

```

# cd /var/lib
# mkdir spamassassin
# touch ./spamassassin/auto-whitelist

```

```
# chown -R sa-milt.sa-milt spamassassin
```

Uma vez finalizada a configuração, vamos iniciar os serviços spamassassin e spamass-milter, além de reiniciar o serviço sendmail.

```
# service spamassassin start
# service spamass-milter start
# service sendmail restart
```

Para verificar possíveis erros, podemos usar o arquivo de log /var/log/maillog.

**Obs.:** Se for verificado o erro:

```
spamd[2138]: auto-whitelist: open of auto-whitelist file failed: auto-
whitelist: cannot open auto_whitelist_path /var/lib/spamassassin/auto-
whitelist: Inappropriate ioctl for device
```

Significa que o arquivo /var/lib/spamassassin/auto-whitelist pode estar com permissões de acesso erradas, onde para corrigir simplesmente altere o dono e grupo do arquivo para sa-milt. Outro ponto importante seria as permissões do diretório /var/lib/spamassassin, que deve ter as mesmas permissões que o arquivo.

## Configurando o Razor

O **Razor** simplesmente ajuda o **SpamAssassin** melhorando a identificação das mensagens. Para instalar, podemos utilizar o comando yum, como mostrado abaixo:

```
# yum install perl-Razor-Agent
```

**Obs.:** Não é necessário dizer que devemos ter acesso a internet para usar o comando yum, senão é preciso instalar os pacotes a partir dos CDs do Fedora Core 6.

Em seguida, é preciso realizar a configuração, conforme apresentada abaixo:

```
# razor-admin -d -create
# razor-admin -register
```

**Obs.:** Os comandos acima realizam uma comunicação com os servidores **Razors**, criando uma identidade em seu servidor. Mensagens do tipo razor-admin finished successfully e Register successful devem ser visualizadas ao final de cada comando.

**Obs.:** Caso mensagens de erro sejam apresentadas, verifique se a porta 2703 (TCP) está liberada em seu Firewall.

Após os comandos bem sucedidos, será criado o diretório /root/.razor, que deve ser copiado para o diretório /var/lib/spamassassin.

**Obs.:** Outro diretório também pode ser utilizado se preferir.

```
# cd /root
# cp -pr .razor /var/lib/spamassassin
```

Devemos alterar o arquivo `/var/lib/spamassassin/.razor/razor-agent.conf` conforme abaixo:

```
debuglevel = 0
razorhome = /var/lib/spamassassin/.razor
```

É preciso reconfigurar o **Razor**, devido sua nova localização, através do comando abaixo:

```
# razor-admin -d -create -home=/var/lib/spamassassin/.razor
```

**Obs.:** Caso nenhum erro ocorra, uma mensagem do tipo `razor-admin finished successfully` deve ser visualizada ao final do comando.

Uma vez finalizada a configuração, vamos adicionar os novos parâmetros na configuração do **SpamAssassin**, através do arquivo `/etc/mail/spamassassin/local.cf`:

```
# cd /etc/mail/spamassassin
# vi local.cf
(Adicionar/Alterar os parâmetros abaixo)
use_razor2 1
razor_config /var/lib/spamassassin/.razor/razor-agent.conf
```

Por último devemos reiniciar o serviço `spamassassin`:

```
# service spamassassin restart
```

## Integrando o ClamAV

### Instalação

Para instalar, utilizaremos o comando `yum`, como mostrado abaixo:

```
# yum install clamav-milter clamav-update
```

**Obs.:** Não é necessário dizer que devemos ter acesso a internet para usar o comando `yum`, senão é preciso instalar os pacotes a partir dos CDs do Fedora Core 6.

### Configuração

Para realizar a integração entre o **ClamAV** e o **Sendmail**, devemos acrescentar ao arquivo `/etc/mail/sendmail.mc` os seguintes parâmetros:

```
INPUT_MAIL_FILTER(`clamav', `S=local:/var/run/clamav-milter/clamav.sock,
F=, T=S:4m;R:4m')dnl
```

**Obs.:** As informações acima foram retiradas do arquivo `INSTALL`, localizado no diretório `/usr/share/doc/clamav-milter-0.88.7`.

Outro arquivo importante na configuração do **ClamAV**, na verdade do **ClamAV Milter**, está localizado no diretório `/etc/sysconfig` com o nome `clamav-milter`. É importante que os arquivos utilizados para a comunicação *socket* sejam os mesmo, onde no nosso caso é `/var/run/clamav-milter/clamav.sock`.

Abaixo temos o arquivo `/etc/sysconfig/clamav-milter`:

```
## The '-blo' options might be usefully here -- especially for testing; see
## "man 8 clamav-milter" for further options
CLAMAV_FLAGS='--max-children=2 -c /etc/clamd.d/milter.conf
local:/var/run/clamav-milter/clamav.sock --force-scan'
CLAMAV_USER='clamilt'
```

**Obs.:** Para maiores detalhes sobre estes e outros parâmetros, seria interessante consultar o manual do **ClamAV Milter**.

```
# man clamav-milter
```

Finalizando a configuração do **ClamAV Milter**, devemos alterar o arquivo `/etc/clamd.d/milter.conf`, conforme apresentado abaixo, onde estão somente as diferenças em relação com o arquivo padrão:

```
# Example
LogSyslog
LogFacility LOG_MAIL
PidFile /var/run/clamav-milter/clamd.pid
# LocalSocket /var/run/clamd/clamd.sock
# FixStaleSocket
```

Finalizando, vamos configurar o **ClamAV Update**, responsável por manter a base de dados de vírus sempre atualizada. O arquivo de configuração é `/etc/freshclam.conf`, onde abaixo temos as alterações realizadas, onde estão apresentadas somente as diferenças em relação ao padrão:

```
#Example
LogSyslog
LogFacility LOG_MAIL
DatabaseMirror db.BR.clamav.net
```

Outro arquivo importante é `/etc/sysconfig/freshclam`, que deve ser alterado, conforme apresentado abaixo, pois por padrão a atualização da base fica desabilitada:

```
## When changing the periodicity of freshclam runs in the crontab,
## this value must be adjusted also. Its value is the timespan between
## two subsequent freshclam runs in minutes. E.g. for the default
##
## | 0 */3 * * * ...
##
## crontab line, the value is 180 (minutes).
# FRESHCLAM_MOD=

## A predefined value for the delay in seconds. By default, the value is
## calculated by the 'hostid' program. This predefined value guarantees
## constant timespans of 3 hours between two subsequent freshclam runs.
##
## This option accepts two special values:
## 'disabled-warn' ... disables the automatic freshclam update and
##                   gives out a warning
## 'disabled'      ... disables the automatic freshclam silently
# FRESHCLAM_DELAY=

### !!!!! REMOVE ME !!!!!
### REMOVE ME: By default, the freshclam update is disabled to avoid
### REMOVE ME: network access without prior activation
# FRESHCLAM_DELAY=disabled-warn # REMOVE ME
```

**Obs.:** Para que a atualização do **ClamAV** funcione é preciso também que o agendador de tarefas, no nosso caso o **cron**, esteja funcionando.

Com as configurações finalizadas, devemos iniciar o serviço **ClamAV Milter** e em seguida reiniciar o **Sendmail**.

```
# service clamav-milter start
# service sendmail restart
```

Com o sistema funcionando corretamente, que pode ser verificado através do arquivo `/var/log/maillog`, é necessário configurar o serviço para que o mesmo seja sempre iniciado junto com o sistema operacional. Para isto, podemos usar o comando apresentado abaixo:

```
# chkconfig clamav-milter on
```

## Monitorando as Mensagens com o Mailgraph

O **Mailgraph** é uma ferramenta simples e prática para gerar gráficos a partir dos logs do MTA, *Mail Transport Agent*, com incrível riqueza de detalhes, facilita o processo, não apenas da distribuição de mensagens, mas também dos erros, vírus e spam.

Para instalar, primeiro iremos resolver as dependências do **Mailgraph**, que se encontram todas já em formato RPM, disponíveis para o Fedora Core 6.

Utilizaremos o comando `yum`, conforme abaixo:

```
# yum install rrdtool rrdtool-perl perl-RRD-Simple perl-File-Tail
```

**Obs.:** Não é necessário dizer que devemos ter acesso a internet para usar o comando `yum`, senão é preciso instalar os pacotes a partir dos CDs do Fedora Core 6.

Uma vez instaladas as dependências, vamos instalar o **Mailgraph**, seguindo o procedimento abaixo:

```
# cd /root
# wget http://people.ee.ethz.ch/~dws/software/mailgraph/pub/mailgraph-1.12.tar.gz
# tar -zxvf mailgraph-1.12.tar.gz
# cd mailgraph-1.12
# cp -p mailgraph.pl /usr/local/bin
# cp -p mailgraph.cgi /var/www/cgi-bin
# cp -p mailgraph-init /etc/init.d/mailgraph
# chkconfig --add mailgraph
# mkdir /var/lib/mailgraph
```

Em seguida, vamos realizar a configuração, alterando no arquivo `/etc/init.d/mailgraph` os seguintes parâmetros:

```
MAIL_LOG=/var/log/maillog
RRD_DIR=/var/lib/mailgraph
```

Também devemos alterar no arquivo `/var/www/cgi-bin/mailgraph.cgi` os parâmetros:

```
my $rrd = '/var/lib/mailgraph/mailgraph.rrd';
my $rrd_virus = '/var/lib/mailgraph/mailgraph_virus.rrd';
```

Com a configuração finalizada, vamos iniciar o serviço `mailgraph` e caso não exista um servidor **Apache** iniciado, também devemos instalar e configurar um, o que não será mostrado neste documento:

```
# service mailgraph start
```

Para visualizar as estatísticas, basta acessar o endereço:

```
http://servidor.seudominio.com.br/cgi-bin/mailgraph.cgi
```

## Referências

- ✓ Sendmail Home Page  
<http://www.sendmail.org>
- ✓ Installing and Configuring the Cyrus IMAP Server  
<http://cyrusimap.web.cmu.edu/imapd/install.html>
- ✓ Sendmail-SMTP-AUTH-TLS-Howto  
[http://www.falkotimme.com/howtos/sendmail\\_smtp\\_auth\\_tls/index.php](http://www.falkotimme.com/howtos/sendmail_smtp_auth_tls/index.php)
- ✓ Solução completa com o Sendmail  
<http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1888>
- ✓ Configurando o SendMail  
<http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=224>
- ✓ Welcome to SpamAssassin  
<http://www.spamassassin.org>
- ✓ SpamAssassin – TrustPath  
<http://wiki.apache.org/spamassassin/TrustPath>
- ✓ SpamAssassin – TrustedRelays  
<http://wiki.apache.org/spamassassin/TrustedRelays>
- ✓ SpamAssassin Configuration Generator  
<http://www.yrex.com/spam/spamconfig.php>
- ✓ Mailgraph - a RRDtool frontend for Mail statistics  
<http://people.ee.ethz.ch/~dws/software/mailgraph>
- ✓ Mail Server Performance Monitoring with Mailgraph  
<http://www.onlamp.com/pub/a/onlamp/2004/08/12/mailgraph.html>
- ✓ How to install Postfix, Amavisd-new, SpamAssassin, Pyzor, Razor, DCC, and ClamAV on Fedora Core 4 – v2.1.8  
<http://www.freespamfilter.org/FC4.html>
- ✓ Clam AntiVirus  
<http://www.clamav.net>