

Servidor de Emails

- Courier IMAP
- Caixas postais em formato Maildir
- SMTP Scao autenticado
- Mysql
- Quota de caixa postal e número de emails enviados
- Proteção SPF e Greylist
- Proteção com antivírus Clamav

Meu nome é Lucas, sou programador e autor do projeto MTA Scao, um servidor SMTP para Linux escrito do zero em C e Assembly.

Comecei a muito tempo um programa anti-spam e isso me jogou dentro do mundo dos emails. Senti então a necessidade de um servidor MTA mais flexível onde eu pudesse acionar programas externos com facilidade e que trabalhasse com esquemas de condições e ações. Nasceu então o projeto.

Vou demonstrar neste tutorial como criar um servidor de emails com domínios virtuais em base de dados Mysql.

O Servidor SMTP será o MTA Scao que irá fazer consultas via modulo externo em um banco de dados Mysql, controlando quota de tamanho de caixa postal e limite de número de emails que cada cliente pode enviar por dia/semana/mês e vamos usar proteção anti-spam com SPF e Greylist e proteção contra vírus usando o Clamav. Para serviços POP3 e IMAP, usaremos o Courier-IMAP que trabalhará com a mesma base de dados usada pelo servidor SMTP.

Versões utilizadas:

- * Linux Slackware 12.0 Full
- * Courier authentication library 0.61.0
- * Courier-IMAP 4.4.1
- * Courier Maildrop 2.0.4
- * MTA Scao 0.28b
- * Modulo externo Scao_mysql 1.2
- * Utilitário Scao_greylist
- * Utilitário Scao_rquota
- * Clamav 0.94

Crie um diretório temporário (/home/temp), faça download dos seguintes arquivos e jogue-os la dentro:

POP3 e IMAP

Courier authentication library:

<http://prdownloads.sourceforge.net/courier/courier-authlib-0.61.0.tar.bz2>

Courier-imap para os serviços de POP3 + IMAP:

<http://prdownloads.sourceforge.net/courier/courier-imap-4.4.1.tar.bz2>

Courier Maildrop (MDA - Mail Delivery Agent)

<http://prdownloads.sourceforge.net/courier/maildrop-2.0.4.tar.bz2>

SMTP

MTA Sceo para o protocolo SMTP:

http://sceo.hospedaria.com.br/downloads/sceo_0_28b.tar.bz2

Modulo Sceo_mysql para permitir que o MTA Sceo trabalhe com base de dados Mysql:

http://sceo.hospedaria.com.br/ferramentas/sceo_mysql_1.2.tar.bz2

Utilitário externo Sceo_greylis:

http://sceo.hospedaria.com.br/ferramentas/sceo_greylis.tar.bz2

Utilitário externo Sceo_rquota para controlar o numero de emails enviados por cliente:

http://sceo.hospedaria.com.br/ferramentas/sceo_rquota.tar.bz2

Antivírus Clamav

Antivírus Clamav para Linux:

<http://ufpr.dl.sourceforge.net/sourceforge/clamav/clamav-0.94.tar.gz>

O primeiro passo é criar o usuário 'sceo' que será utilizado pelo Courier e Sceo nas caixas postais dos clientes:

```
# useradd -u 105 sceo
```

Utilize o UID 105 que é o que usaremos no banco de dados. Se quiser você pode usar outro mas lembre-se de alterar o UID em todas as etapas abaixo

Preparando o Banco de dados Mysql:

No Slackware, é preciso criar o banco de dados básico de funcionamento do Mysql, sem isso ele não liga.

Digite os seguintes comandos:

```
# /usr/bin/mysql_install_db
```

```
# chown -R mysql. /var/lib/mysql/mysql
```

```
# chown -R mysql. /var/lib/mysql/test
```

```
# chmod 555 /etc/rc.d/rc.mysql
```

Ligue o Mysql:

```
# /etc/rc.d/rc.mysql start
```

Vamos criar o banco de dados:

Digite os seguintes comandos:

```
# mysql
```

```
mysql> create database mail;
```

```
mysql> grant all privileges on mail.* to sceo@localhost identified by
"minhasenha" with grant option;
```

```
mysql> flush privileges;
```

```
mysql> quit
```

Lembre-se de colocar uma outra senha qualquer no comando ...identified by "minhasenha"... acima

Pronto, o banco de dados esta criado, vamos criar agora as tabelas. Crie um arquivo chamado mail.sql (# pico /etc/mail.sql) e coloque o seguinte texto dentro dele:

```
-----
-- Estrutura da tabela `aliases`
--
CREATE TABLE `aliases` (
  `alias` varchar(255) NOT NULL default '',
  `rcpt` varchar(255) default NULL,
  KEY `alias` (`alias`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
-----
-- Estrutura da tabela `domain`
--
CREATE TABLE `domain` (
  `name` char(128) NOT NULL default '',
  PRIMARY KEY (`name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
-----
-- Estrutura da tabela `users`
--
CREATE TABLE `users` (
  `id` int(32) unsigned NOT NULL auto_increment,
  `mail` char(128) NOT NULL default '',
  `pass` char(128) default NULL,
  `uid` int(10) unsigned default '105',
  `gid` int(10) unsigned default '100',
  `home` char(255) default NULL,
  `maildir` char(255) default NULL,
  `date_add` date default NULL,
  `time_add` time default NULL,
  `domain` char(128) default NULL,
  `name` char(255) default NULL,
  `ok` tinyint(3) unsigned default '1',
  `quota` char(255) default '250000000',
  PRIMARY KEY (`mail`),
  KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
-----
-- Fim do arquivo mail.sql
```

250000000 equivale a aproximadamente 250MB para cada caixa postal. Você pode alterar este valor se quiser. Em vários tutoriais por ai, você pode notar um 'S' no final do valor da quota (Ex: 250000000S), ele serve para fazer o maildrop controlar a quota pra você, mas NÃO é isso que queremos. O Sced é que vai controlar e você NÃO deve colocar o 'S' no final.

Agora execute o comando:

```
# mysql mail < /etc/mail.sql
```

Se não houve mensagem alguma de resposta, tudo ocorreu bem, o banco de dados esta criado e suas tabelas também (Assim espero)

Entre no Mysql e insira alguns registros que usaremos para testes.

Vamos usar um domínio de testes chamado dominio.com.br com a conta de email lucas@dominio.com.br:

```
# mysql
```

```
mysql> use mail;
```

```
mysql> INSERT INTO domain VALUES ('dominio.com.br');
```

```
mysql> INSERT INTO users ( mail,home,pass,maildir,date_add,time_add,domain,name )  
VALUES ( 'lucas@dominio.com.br', '/home/mail/', encrypt('mudar123'),  
'dominio.com.br/lucas/Maildir/', '2008-08-14', '00:00:00', 'dominio.com.br',  
'Lucas Teste' );
```

```
mysql> quit
```

Ok, a parte do Mysql esta finalizada.

Vamos agora a parte mais complicada que é a instalação do Courier-IMAP, sua lib de autenticação e o Maildrop.

Courier-Authlib

Instalando o Courier authentication library.

```
# cd /home/temp
```

O processo de 'configure' é demorado, não se assuste. Parece que ele entra em loop infinito mas é apenas uma ilusão.

```
# tar xvjf courier-authlib-0.61.0.tar.bz2
```

```
# cd courier-authlib-0.61.0
```

```
# ./configure --sysconfdir=/etc/courier
```

```
# make
```

```
# make install
```

```
# make install-configure
```

Copie as libs que o instalador jogou em /usr/local/lib/courier-authlib para /usr/lib para não termos problemas para compilar o Courier-IMAP.

```
# cp /usr/local/lib/courier-authlib/* /usr/lib
```

O próximo passo é configurar o authdaemon:

Abra o arquivo /etc/courier/authlib/authdaemonrc e edite as opções **authmodulelist** e **authmodulelistorig** de forma a ficarem apenas com o modulo Mysql

```
# pico /etc/courier/authlib/authdaemonrc
```

```
authmodulelist="authmysql"  
authmodulelistorig="authmysql"
```

Agora limpe o arquivo /etc/courier/authlib/authmysqlrc e insira nele as informações descritas mais abaixo.

```
# rm /etc/courier/authlib/authmysqlrc
```

```
# pico /etc/courier/authlib/authmysqlrc
```

E deixe-o conforme as opções abaixo:

```
MYSQL_SERVER          localhost  
MYSQL_USERNAME        sceo  
MYSQL_PASSWORD        minhasenha  
MYSQL_PORT            3306  
MYSQL_OPT              0  
MYSQL_DATABASE        mail  
MYSQL_USER_TABLE      users  
MYSQL_CRYPT_PWFIELD   pass  
MYSQL_UID_FIELD       uid  
MYSQL_GID_FIELD       gid  
MYSQL_LOGIN_FIELD     mail  
MYSQL_HOME_FIELD      home  
MYSQL_NAME_FIELD      name  
MYSQL_MAILDIR_FIELD   maildir  
MYSQL_QUOTA_FIELD     quota  
MYSQL_WHERE_CLAUSE    ok=1
```

Obs: Lembre-se de colocar a senha correta em MYSQL_PASSWORD. É a mesma senha que você usou para criar o banco de dados mais acima.

Courier-IMAP

```
# cd /home/temp
```

```
# tar xvjf courier-imap-4.4.1.tar.bz2
```

Para compilar o Courier-IMAP, é preciso não ter privilégios de root. Mas antes vamos alterar as permissões dos arquivos do Courier para o usuário que vai utilizar para compila-lo. No meu caso é o usuário lucas:

```
# useradd lucas
```

```
# passwd lucas
```

```
<Entre com a senha do 'lucas'>
```

```
# chown -R lucas.courier-imap-4.4.1
```

Logue-se com o usuário comum ('lucas' no meu caso) e depois compile o programa. Não se assuste, o processo é meio demorado mesmo e parece ter entrado em loop infinito, é normal.

```
# cd /home/temp/courier-imap-4.4.1
```

```
# ./configure --sysconfdir=/etc/courier
```

```
# make
```

Após compila-lo é preciso entrar com privilégios de root para instalar.

```
# su -
```

```
...entre com a senha do root
```

```
# cd /home/temp/courier-imap-4.4.1
```

```
# make install
```

```
# make install-configure
```

Abaixo vou demonstrar como ligar/desligar o Courier-IMAP, mas você NÃO precisa liga-lo agora. Vou passar mais abaixo um script para controla-lo.

Para iniciar o IMAP e POP3:

```
/usr/lib/courier-imap/libexec/imapd.rc start
```

```
/usr/lib/courier-imap/libexec/pop3d.rc start
```

Para parar o IMAP e POP3:

```
/usr/lib/courier-imap/libexec/imapd.rc stop
```

```
/usr/lib/courier-imap/libexec/pop3d.rc stop
```

Script de controle do Courier-IMAP:
Crie o arquivo /etc/rc.d/rc.courier

```
# pico /etc/rc.d/rc.courier
```

Copie e cole o conteúdo abaixo no arquivo /etc/rc.d/rc.courier

```
#!/bin/sh
```

```
#####
# Script de controle do servidor Courier-IMAP
# ./rc.courier start    -> Ligar servidor
# ./rc.courier stop    -> Parar servidor
# ./rc.courier restart -> Reiniciar servidor
#
#
#
ligar_courier() {
echo "Iniciando Courier-IMAP..."
/usr/local/sbin/authdaemon start
/usr/lib/courier-imap/libexec/imapd.rc start
/usr/lib/courier-imap/libexec/pop3d.rc start
}
# Stop courier:
parar_courier() {
echo "Parando Courier-IMAP..."
/usr/lib/courier-imap/libexec/imapd.rc stop
/usr/lib/courier-imap/libexec/pop3d.rc stop
/usr/local/sbin/authdaemon start
}
case "$1" in
'start')
ligar_courier
;;
'stop')
parar_courier
;;
'restart')
parar_courier
sleep 1
ligar_courier
;;
*)
echo "Os parametros aceitaveis sao: $0 start/stop/restart"
esac
#----- Fim do SCRIPT -----
```

De permissão de execução ao script:

```
# chmod 700 /etc/rc.d/rc.courier
```

Testando o IMAP e POP3:
Ligue o Courier:

```
# /etc/rc.d/rc.courier start
```

Antes de testa-lo precisamos criar o diretório do usuário lucas@dominio.com.br.
No banco de dados dissemos que a caixa postal desta conta esta em:
/home/mail/dominio.com.br/lucas/Maildir

Crie o caminho assim:

```
# mkdir -p /home/mail/dominio.com.br/lucas
```

```
# cd /home/mail/dominio.com.br/lucas
```

```
# /usr/lib/courier-imap/bin/maildirmake Maildir
```

```
# touch /home/mail/dominio.com.br/lucas/Maildir/maildirsiz
```

```
# chown -R sceo. /home/mail
```

ATENÇÃO:

Tenha em mente que para o maildrop entregar a mensagem ele precisa que o diretório 'mail', 'dominio.com.br', e que tudo dentro de 'lucas' esteja com permissão para o usuário 'sceo', senão, você terá o erro 'Home directory owned by wrong user', então, sempre que criar um novo domínio lembre-se de definir o usuário 'sceo' apenas para a pasta do domínio e depois para a subpasta dos emails que criar. Não execute sempre o 'chown -R sceo. /home/mail' a cada conta criada pois se você tiver 1000 contas de emails o sistema operacional vai lembrar muito da sua mamãezinha por ter que redefinir a permissão de milhões de diretórios desnecessariamente. Coloque-se no lugar do processador. ;-)

É extremamente importante que o arquivo maildirsiz esteja sempre com permissão para o usuário 'sceo'. O Courier e o Maildrop atualizam neste arquivo a movimentação dos emails da caixa postal. O modulo sceo_mysql analisa este arquivo para saber se a caixa postal esta lotada ou não. O Modulo só considera a caixa postal lotada depois que ela excedeu seu limite.

Testando o IMAP:

```
# telnet localhost 143
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.  
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT  
THREAD=REFERENCES SORT QUOTA IDLE ACL ACL2=UNION STARTTLS] Courier-IMAP ready.  
Copyright 1998-2008 Double Precision, Inc. See COPYING for distribution  
information.
```

Digite:

```
? login lucas@dominio.com.br mudar123
```

Se tudo ocorreu bem, ele responderá:

```
? OK LOGIN Ok.
```

Para sair:

```
? logout
```

Testando o POP3:

```
# telnet localhost 110
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.  
+OK Hello there.
```

Digite:

```
user lucas@dominio.com.br
```

```
+OK Password required.
```



```
pass mudar123
+OK logged in.
```

```
quit
+OK Bye-bye.
```

Maildrop

O Maildrop é o programa que recebe o email do Sceo e entrega-o em sua respectiva caixa postal.

```
# cd /home/temp
# tar xvjf maildrop-2.0.4.tar.bz2
# cd maildrop-2.0.4
# ./configure --sysconfdir=/etc/courier --enable-maildirquota
# make
# make install-strip
# make install-man
```

É preciso setar o uidbit do maildrop para ele conseguir entregar as mensagens.

```
# chmod 4555 /usr/local/bin/maildrop
```

Feito, Maildrop instalado e terminamos a parte do Courier

MTA Sceo

A instalação dele é extremamente simples, copie o arquivo sceo_0_28b.tar.bz2 para a raiz do sistema e descompacte-o lá:

```
# cd /home/temp
# cp sceo_0_28b.tar.bz2 /
# cd /
# tar xjvf sceo_0_28b.tar.bz2
```

Edite seu arquivo de configuração:

```
# pico /usr/local/sceo/sceo.conf
```

Em **Server_name** coloque o nome do endereço reverso de seu servidor.
Altere seu arquivo sceo.conf de forma a ficar com estas opções:

```
Module "/usr/local/sceo/mod/sceo_mysql /usr/local/sceo/mod/sceo_mysql.conf"  
Dlocal "/usr/local/bin/maildrop -d %l <"
```

Lembre-se de **COMENTAR** ou **APAGAR** as antigas linhas das opções alteradas:

```
Module "" e Dlocal "/usr/bin/procmail -d %l <"
```

Salve e saia do arquivo de configuração.

Faça um link para o programa 'fila' do /usr/local/sceo/fila em /usr/sbin. Este programa é usado listar a atual fila do Sceo.

```
# ln -s /usr/local/sceo/fila /usr/sbin/fila
```

Vamos instalar o modulo sceo_mysql:

Copie o sceo_mysql_1.1.tar.bz2 para /usr/local/sceo/mod e descompacte-o lá:

```
# cd /home/temp
```

```
# cp sceo_mysql_1.2.tar.bz2 /usr/local/sceo/mod
```

```
# cd /usr/local/sceo/mod
```

```
# tar xjvf sceo_mysql_1.2.tar.bz2
```

Agora vamos compila-lo:

```
# gcc sceo_mysql.c -o sceo_mysql -lmysqlclient -lcrypt
```

Lembre-se que estou instalando os programas em um Slackware 12 Full. Caso de algum erro na compilação, com certeza é porque você não instalou todos os pacotes do mysql.

Edite o arquivo sceo_mysql.conf:

```
# pico sceo_mysql.conf
```

Deixe-o desta forma:

```
host= localhost  
db= mail  
user= sceo  
pass= minhasenha  
user_table= users  
user_login_field= mail  
user_pass_field= pass  
user_pass_type= crypt  
user_home_field= home  
user_maildir_field= maildir  
user_quota_field= quota  
quota_type= maildirsiz  
domain_table= domain  
domain_field= name  
alias_table= aliases  
alias_field= alias  
alias_rcpt_field= rcpt
```

Lembre-se de deixar a opção pass= com a senha correta de seu banco de dados. Salve e saia do arquivo.

Vamos testar o modulo:

```
# ./sceo_mysql sceo_mysql.conf
+OK SCEO_Mysql v 1.1 Conectado
```

```
local dominio.com.br
+OK
```

```
local teste.com.br
-Nao local
```

```
user lucas@dominio.com.br
+OK
```

```
user teste@dominio.com.br
-Email inexistente
```

```
quota lucas@dominio.com.br
+OK 250000000
```

Tudo funcionando, digite CTRL + C para sair do modulo.

250000000 (~ 250 Mb) é o valor que especificamos como padrão na criação do banco de dados

Agora vamos ligar o Sceo:

```
# /etc/rc.d/rc.sceo start
```

De um ps ax para ver se ele apareceu na lista de processos com seus módulos:

```
# ps ax
```

```
.
.
.
6222 ?      Ss      0:00 /usr/local/sceo/sceo
6223 ?      S       0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
6225 ?      S       0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
6227 ?      S       0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
6229 ?      S       0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
6231 ?      S       0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
6233 ?      S       0:00 /usr/local/sceo/sceo
```

Os módulos e o processo principal rodam como root. Mas os processos filhos criados para tratar as novas conexões e os arquivos de fila rodam com usuário 'sceo'

Vamos testa-lo:

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 localhost SMTP SCEO v0.28b
```

```
ehlo testador
250-localhost
250-SIZE 10485760
250-AUTH PLAIN LOGIN
250 8BITMIME
```

```
mail from: <>
250 Remetente liberado
```

```
rcpt to: <lucas@dominio.com.br>
250 Destinatario liberado
```

Teste agora um usuário que NAO existe:

```
rcpt to: <teste@dominio.com.br>
550 Esse email nao existe. (no such user) <teste@dominio.com.br>
```

```
quit
```

Se você quer que o Sceo reenvie os emails de sua rede interna (Relay) basta acrescentar a seguinte linha no arquivo /usr/local/sceo/regras_conexao. Digamos que sua rede interna seja 10.0.0.0/24:

```
Ip? "10.0.0.0/24" Relay!
```

Salve e saia do arquivo. Não é preciso reiniciar o Sceo

Tudo ok com o servidor SMTP.

Configurando o SPF

O SPF é fácil. O Sceo tem suporte nativo a SPF. Edite o arquivo /usr/local/sceo/regras_remetente:

```
# pico /usr/local/sceo/regras_remetente
```

Acrescente as seguintes linhas:

```
Spf_test!
Spf_resp? "4" Reply! "550 Fail, voce nao passou pelo teste SPF" Deny!
```

E se você estiver de mau humor, bloqueie as respostas 3 também:

```
Spf_resp? "3" Reply! "550 Fail, voce nao passou pelo teste SPF" Deny!
```

Salve e saia do arquivo. Não é preciso reiniciar o Sceo.

Tabela de respostas

Resp. do SPF	Descrição
0	- Sem SPF
1	- Passou no teste. (Pass)
2	- O servidor não pertence a rede que mantém o domínio (Neutral)
3	- Este servidor não deveria estar mandando esse email (SoftFail)
4	- Email DEVE ser recusado (Fail)
5	- Erro temporário. (TempError)
6	- Erro permanente durante a checagem do SPF (PermError)

Nos servidores que configurei eu sempre mando o Scao adicionar a resposta SPF no Header do email para uma posterior análise de um programa anti-spam:

```
Hadd! "X-Spf: %y"
```

A regra acima pode ser colocada no arquivo `/usr/local/scao/regras_remetente` logo após as regras de `Spf_resp?` ...

Greylist

Lógica da Greylist:

A Greylist faz com que um email seja temporariamente recusado por nosso servidor, se por acaso o Host remoto seguir as regras do protocolo SMTP ele vai tentar reenviar a mensagem mais tarde onde será aceita sem problemas pelo nosso servidor. Grande parte dos Spammers não tentam reenviar uma mensagem em caso de erro temporário. Todo esse processo é controlado via banco de dados pelo programa `scao_greylist`.

O filtro Greylist é muito poderoso e barra muito lixo.

Muitos servidores por aí usam Greylist baseada no IP do servidor remoto, mas há casos em que um Host possui vários IPs e isso prejudica muito o esquema básico da Greylist. Vamos fazer então o `scao_greylist` trabalhar não com o IP mas com o endereço reverso do servidor e caso o mesmo não tenha, o Scao automaticamente utilizará o endereço IP. O Gmail é um exemplo onde um mesmo endereço reverso tenta entregar uma mensagem com muitos IPs diferentes.

Outro diferencial é que NÃO vamos rodar Greylist caso o SPF do domínio esteja OK (Pass). Isto agiliza a entrega de emails de servidores válidos (vindos do Gmail, Hotmail, etc...). Isto me tirou muita dor de cabeça já que o spammer que é capaz de configurar um registro SPF, consegue passar tranquilamente por uma Greylist.

Para tudo ficar organizado crie o diretório `/usr/local/scao/uteis`

```
# mkdir /usr/local/scao/uteis
```

```
# cd /home/temp
```

```
# cp scao_greylist.tar.bz2 /usr/local/scao/uteis/
```

```
# cd /usr/local/scao/uteis
```

```
# tar xjvf scao_greylist.tar.bz2
```

Crie o banco de dados

```
# mysql
```

```
mysql> create database sceo_greylis;
```

```
mysql> grant all privileges on sceo_greylis.* to sceo@localhost identified by "minhasenha" with grant option;
```

```
mysql> flush privileges;
```

```
mysql> quit
```

Você pode usar outro usuário e senha se quiser.

Jogue do arquivo sceo_greylis.sql para o banco de dados:

```
# mysql sceo_greylis < sceo_greylis.sql
```

Agora é só editar o sceo_greylis.c:

```
# pico sceo_greylis.c
```

Procure as defines USUARIO e SENHA para mudar seus 'valores' se necessário

```
#define USUARIO "sceo"  
#define SENHA "minhasenha"
```

Lembre-se de colocar a senha que especificou na criação do banco de dados sceo_greylis acima.

Salve, saia do arquivo e compile-o:

```
# gcc sceo_greylis.c -o sceo_greylis -lmysqlclient
```

Caso de algum erro para compilar, deve ser porque você não tem todos os pacotes do Mysql instalado.

Precisamos configurar o programa sceo_greylis para o Sceo chama-lo

Copie o binário gerado para um diretório que esteja no PATH (Ex: /usr/bin)

```
# cp sceo_greylis /usr/bin
```

Edite o arquivo /usr/local/sceo/regras_destinatario

```
# pico /usr/local/sceo/regras_destinatario
```

Acrescente a seguinte linha:

```
!Spf_resp? "1" !Auth? Internal_rcpt? Exec! "sceo_greylis -i %s -f %f -t %r"  
Exec_resp? "1" Reply! "451 Tente mais tarde" Deny!
```

OBS: Todos comandos acima DEVEM ficar em uma única linha.

Esta linha roda o programa externo sceo_greylis caso a resposta do SPF NÃO seja 1 e o cliente NÃO seja autenticado. A exclamação(!) na frente da condição funciona como um negador.

Salve e saia do arquivo. Não é preciso reiniciar o Sceo

Você só precisa reinicia-lo quando alterar o arquivo sceo.conf

Agende o seu CRON para rodar a seguinte linha uma vez ao dia:

```
sceo_greylist -clean
```

Linha do CRON:

```
00 0 * * * /usr/bin/sceo_greylist -clean
```

Isso fará a exclusão de registros mortos mantendo o banco de dados limpo e rápido

Quota de caixa postal

Para controlar a quota, você não precisa fazer nada, o Sceo automaticamente controla a quota da caixa postal para você baseado no que esta no banco de dados. Ele faz isso perguntando ao modulo externo sceo_mysql que esta preparado para testar o atual tamanho de uma caixa postal maildir do Courier.

Ele le o arquivo maildirsize da caixa postal e ao mesmo tempo resume as informações la contidas para não deixar o arquivo muito grande.

O Sceo só recusa o email se o limite já estiver estourado.

Número de emails que um cliente pode enviar

Neste exemplo vou definir um limite de 1000 emails por dia por cliente.

Copie o sceo_rquota.tar.bz2 para /usr/local/sceo/uteis

```
# cd /home/temp
```

```
# cp sceo_rquota.tar.bz2 /usr/local/sceo/uteis
```

```
# cd /usr/local/sceo/uteis
```

```
# tar xjvf sceo_rquota.tar.bz2
```

Crie o banco de dados:

```
# mysql
```

```
mysql> create database sceo_rquota;
```

```
mysql> use sceo_rquota;
```

```
mysql> grant all privileges on sceo_rquota.* to sceo@localhost identified by "minhasenha" with grant option;
```

```
mysql> flush privileges;
```

```
mysql> quit
```

Faca o dump da tabela no banco de dados:

```
# mysql sceo_rquota < sceo_rquota.sql
```

Edite o arquivo sceo_rquota.c e procure as defines USER E PASS para mudar seus parâmetros se necessário:

```
# pico sceo_rquota.c
```

```
#define USER "sceo"
```

```
#define PASS "minhasenha"
```

Lembre-se do detalhe da senha, use a que você especificou na criação do banco de dados sceo_rquota acima.

Salve e saia do arquivo.

Compile o programa:

```
# gcc sceo_rquota.c -o sceo_rquota -lmysqlclient
```

Copie o binário gerado para um diretório que esteja no PATH (Ex: /usr/bin)

```
# cp sceo_rquota /usr/bin
```

Abra o arquivo de regras /usr/local/sceo/regras_destinatario

```
# pico /usr/local/sceo/regras_destinatario
```

Acrescente a seguinte linha logo no início do arquivo:

```
Auth? !From? "" Exec! "/usr/bin/sceo_rquota -c -u %f" Exec_resp? "1" Reply! "500  
Limite de envio de emails foi excedido" Deny!
```

OBS: Os comandos acima DEVEM ficar na mesma linha.

Salve e saia do arquivo

Abra o arquivo de regras /usr/local/sceo/regras_data

```
# pico /usr/local/sceo/regras_data
```

Acrescente a seguinte linha logo no início do arquivo:

```
Auth? !From? "" Exec! "/usr/bin/sceo_rquota -a %n -u %f -d 1000"
```

Agende seu cron para rodar o sceo_rquota de forma a zerar o número de destinatários enviados de cada cliente.

Você pode agendar o cron uma vez ao dia, semana, mês ou da forma que quiser.

Exemplos de configurações no Cron.

Por dia:

```
00 0 * * * /usr/bin/sceo_rquota -z
```

Por semana:

```
00 0 * * 0 /usr/bin/sceo_rquota -z
```

Por mês:

```
00 0 1 * * /usr/bin/sceo_rquota -z
```


Antivírus Clamav:

O Clamav é um projeto free e a atual versão suporta analisar arquivos de emails diretamente. Ele consegue procurar por seus anexos e scannea-los.

```
# cd /home/temp
# tar xzvf clamav-0.94.tar.gz
# groupadd clamav
# useradd -g clamav -s /bin/false -c "Clam AntiVirus" clamav
# cd clamav-0.94
# ./configure --sysconfdir=/etc --libdir=/usr/lib
# make
# make install
```

Agora edite o arquivo /etc/clamd.conf e comente a linha 'Example' com um '#':
#Example

Faça o mesmo com o arquivo de configuração do programa de update freshclam:
/etc/freshclam.conf

Agende seu Cron para atualizar o banco de dados do Clamav uma vez ao dia:
00 0 * * * /usr/local/bin/freshclam --daemon-notify=/etc/clamd.conf

Ligue o clamd com o comando:
/usr/local/sbin/clamd

Vamos fazer o Sceo acionar o Clamav a cada email que chegar.
Edite o arquivo /usr/local/sceo/regras_data
pico /usr/local/sceo/regras_data

Acrescente a seguinte linha:

```
Exec! "/usr/local/bin/clamscan --quiet /var/spool/sceo/c%d" Exec_resp? "1"  
Reply! "500 Email com VIRUS" Deny!
```

OBS: Os comandos acima DEVEM ficar na mesma linha.

Salve e saia do arquivo
Não é preciso reiniciar o Sceo

Sempre que um email for recebido pelo Sceo, ele manda o clamscan verificar o email que esta ainda na fila e se a resposta do clamscan for "1", significa que o email esta infectado e ele é recusado antes de ser aceito por completo.

Se quiser, você pode mandar o Sceo gravar um Log, basta usar a seguinte linha no lugar da regra anterior:

```
Exec! "/usr/local/bin/clamscan --quiet /var/spool/sceo/c%d" Exec_resp? "1"  
Reply! "500 Email com VIRUS" Deny! Log! "/var/log/sceo/virus.log:ID %d recusado  
por conter virus"
```

OBS: Os comandos acima DEVEM ficar na mesma linha.

Edite o seu rc.local para ligar automaticamente os servidores.
pico /etc/rc.d/rc.local

Acrescente as seguintes linhas no arquivo:

```
### Ligando Antivírus ###  
/usr/local/sbin/clamd
```

```
### Ligando o MTA Sceo ###  
/etc/rc.d/rc.sceo start
```

```
### Ligando o Courier-IMAP ###  
/etc/rc.d/rc.courier start
```

Seu servidor de emails esta terminado.

Tentei ser o mais claro possível sem estender muito o tutorial. Espero ter ajudado.

Você pode consultar uma lista de condições e ações do MTA Sceo para um maior controle de seu servidor de emails em:

<http://sceo.hospedaria.com.br/documentacao.html>

Lucas Priori
lpriori@hospedaria.com.br

Sites dos projetos e referencias:

MTA Sceo:
<http://sceo.hospedaria.com.br>

Courier:
<http://www.courier-mta.org>

Clamav:
<http://www.clamav.net>