Servidor de E-mails com MTA Brasileiro

- Servidor dos protocolos SMTP autenticado, POP3 e IMAP
- Usuários em Mysql
- Quota de caixa postal
- Controle da quantidade de emails que cada conta pode enviar por dia, semana ou mês
- Proteção antispam usando Greylist, SPF, RBL, Spamassassin
- Proteção com o antivírus Clamav
- Técnicas para corte de Spam via regras do Sceo incluindo proteção automática contra DoS. O Sceo automaticamente adiciona uma regra no iptables cortando o IP do atacante.

Este tutorial foi escrito para servir de guia para quem quer configurar um servidor de emails completo, rápido e de fácil controle sobre seus emails. Este tutorial foi testado, ou seja, eu formatei um micro de testes, instalei o Linux e dei CTRL+C nos comandos passados aqui e CTRL+V em meu terminal e no final eu já tinha o servidor funcionando. Siga este tutorial passo a passo e ele te entregará um servidor de Emails.

Usaremos o Dovecot para os protocolos POP3 e IMAP e para SMTP usaremos o MTA Sceo.

O Sceo é um projeto brasileiro escrito do zero em C e Assembly inicialmente para Linux. Desde que o projeto foi divulgado, eu recebi muitos contatos e entre eles destaca-se Renato Martins e Ananias Filho.
Renato me ajudou muito com o Sceo. Gostaria de agradece-lo pois duas versões saíram com dicas de melhorias e funções que ele solicitou e que achei extremamente válidas para o projeto.
E um agradecimento à Ananias também, que começou a fazer um WebAdmin para as contas de emails, o Sceo-UI, que tornou-se um excelente administrador de emails muito útil e intuitivo.

Vamos à instalação. Versões utilizadas:

- Linux Slackware 12.0 (Full)
- MTA Sceo 0.30
- Dovecot 1.2.14
- Modulo sceo_mysql 1.2.5 parar dar suporte a Mysql ao Sceo
- Utilitário Greylist
- Utilitário Sceo_rquota
- Clamav 0.95.3
- Spamassassin 3.2.5

OBS: *Em todos os exemplos passados aqui eu usei o endereço IP como sendo 240.240.240.1. Este IP não faz parte da rede dos servidores que eu administro e eu nem fiz uma pesquisa para descobrir a quem pertence. É usado apenas como* ***exemplo*** *e você deve ficar atendo no Ctrl+C, Ctrl+V para colocar o IP do seu servidor.*

Crie um diretório temporário para fazermos download dos programas e compila-los:

```
# mkdir /home/progs
```

Fazendo download de todos os programas:

MTA Sceo: (http://www.sceo.com.br)

```
# cd /home/progs
# wget http://www.sceo.com.br/downloads/sceo_0.30.tar.bz2
# wget http://www.sceo.com.br/ferramentas/sceo_mysql_1.2.5.tar.bz2
# wget http://www.sceo.com.br/ferramentas/sceo_greylist.tar.bz2
# wget http://www.sceo.com.br/ferramentas/sceo_rquota.tar.bz2
```

Dovecot: (http://www.dovecot.org)

```
# wget http://www.dovecot.org/releases/1.2/dovecot-1.2.14.tar.gz
```

Clamav: (http://www.clamav.net)

```
# wget http://ufpr.dl.sourceforge.net/project/clamav/clamav/0.95.3/clamav-0.95.3.tar.gz
```

Spamassassin: (http://spamassassin.apache.org)

```
# wget http://ftp.unicamp.br/pub/apache/spamassassin/source/Mail-SpamAssassin-3.2.5.tar.bz2
```

E estes são alguns módulos que o Spamassassin usa. As vezes o link não funciona, então, fique tentando até conseguir.

```
# wget http://search.cpan.org/CPAN/authors/id/G/GA/GAAS/Digest-SHA1-2.12.tar.gz
# wget http://search.cpan.org/CPAN/authors/id/G/GA/GAAS/HTML-Parser-3.65.tar.gz
# wget http://search.cpan.org/CPAN/authors/id/O/OL/OLAF/Net-DNS-0.66.tar.gz
# wget http://search.cpan.org/CPAN/authors/id/M/MI/MIKER/NetAddr-IP-4.027.tar.gz
# wget http://search.cpan.org/CPAN/authors/id/T/TO/TOMHUGHES/IO-Zlib-1.10.tar.gz
```

Iniciando a instalação.

MTA-Sceo

Para instalar o Sceo basta você descompacta-lo na raiz do sistema.

```
# cp sceo_0.30.tar.bz2 /
# cd /
# tar xjf sceo_0.30.tar.bz2
```

Foram criados os seguintes diretórios:

```
/usr/local/sceo
/var/spool/sceo
/var/log/sceo
```

O binário e os arquivos de configuração do Sceo estão em **/usr/local/sceo**. A 'fila' ficará em **/var/spool/sceo** e os logs em **/var/log/sceo**. Foi criado o rc.sceo dentro de /etc/rc.d para ligar/desligar o MTA.

Adicione o usuário 'sceo':

```
# groupadd -g 105 sceo
# useradd -g sceo -u 105 sceo
```

As pastas /var/log/sceo e /var/spool/sceo ficaram com permissão 777 porque não da para saber qual usuário você utilizará com o MTA. Vamos arrumar a permissão para ficar mais seguro:

```
# chown -R sceo. /var/spool/sceo /var/log/sceo
# chmod -R 744 /var/spool/sceo /var/log/sceo
```

Vamos à configuração. (Eu gosto do 'pico' como editor de textos, use o de sua preferência).

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

Procure e altere a opção 'Server_name' colocando o nome reverso do endereço IP de seu servidor (FQDN)

Exemplo:

```
Server_name "mx1.meudominio.com.br"
```

É importantíssimo colocar corretamente o nome do seu servidor na Internet para não ser recusado pelos demais MTAs.

Procure agora a linha ‘Module ""’ e comente-a com ‘#’ na frente. Ela deve ficar assim:

```
#Module ""
```

Descomente a linha Module que ponta para o modulo MySQL. A linha deve ficar assim:

```
Module "/usr/local/sceo/mod/sceo_mysql /usr/local/sceo/mod/sceo_mysql.conf"
```

Agora procure e altere a opção Dlocal para:

```
Dlocal "/usr/local/libexec/dovecot/deliver -d %l <"
```

Salve e feche o sceo.conf.

Vamos criar alguns links.
O sceo vem com as ferramentas sceo_mail e fila.

- ➔ sceo_mail - Substitui o ‘sendmail’ usado pelos programas locais como Cron, PHP (função mail), etc. Vamos linka-lo com o nome /usr/bin/sendmail, assim, você não precisa reconfigurar esses programas.
- ➔ fila - Mostra a atual fila de emails em processamento do Sceo. Ele praticamente lista o /var/spool/sceo

```
# ln -sf /usr/local/sceo/sceo_mail /usr/bin/sendmail
# ln -sf /usr/local/sceo/fila /usr/bin/fila
```

E para o binário do Sceo, para facilitar o processamento manual de um ID de fila com a opçãp -p.

```
# ln -sf /usr/local/sceo/sceo /usr/bin/sceo
```

Em relação a quota, não é preciso fazer nada. Ela funcionará corretamente para cada email com base no que você colocar em seu registro no banco de dados.

Modulo Sceo_MySQL
No Slackware é preciso rodar os seguintes comandos para preparar o MySQL:

```
# /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.mysqld
```

Agora ligue o banco de dados:

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

Vamos criar o banco de dados das contas de e-mails.

```
# mysql
mysql> create database mail;
mysql> grant all privileges on mail.* to sceo@localhost identified by "minhasenha";
mysql> flush privileges;
```

```
mysql> quit;
```

LEMBRETE*: Em todos os comandos e exemplos de configuração abaixo que requerem o uso da senha do banco de dados eu usarei "minhasenha", você deve estar atento para colocar a senha correta. Em um outro tutorial que escrevi, recebi muitos e-mails de pessoas em que o seu servidor não funcionava porque esqueciam de colocar a senha correta.*

Instalando o módulo em seu devido lugar:

```
# cd /usr/local/sceo/mod
# cp /home/progs/sceo_mysql_1.2.5.tar.bz2 .
# tar xjf sceo_mysql_1.2.5.tar.bz2
```

Compile o modulo com o comando:

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

Se a compilação acima der algum erro é porque você não tem o Mysql-devel instalado corretamente ai em seu sistema operacional. Lembre-se que eu estou usando um Linux Full (Tudo instalado).
Agora edite o sceo_mysql.conf

```
# pico sceo_mysql.conf
```

Altere a opção 'Pass=' colocando a senha que usou na criação do banco de dados.
Salve e feche o arquivo. Agora crie a estrutura do banco de dados 'mail' com o seguinte comando:

```
# mysql mail < sceo_mysql.sql
```

No meu caso, acabei de instalar o Slackware e o MySQL por padrão não vem com a senha do root ligada, por isso eu não precisei especificar a senha do root do banco de dados na linha acima. Caso o seu MySQL esteja com senha, use o comando ***"mysql -p mail < sceo_mysql.sql"*** e entre com a senha do root do MySQL quando for solicitado.

Insira um registro de e-mail para testes:

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

O comando abaixo é em uma única linha:

```
mysql> INSERT INTO users (mail,home,pass,maildir,date_add,time_add,domain,name)
VALUES ('lucas@dominio.com.br','/home/mail/dominio.com.br/lucas/',
encrypt('senha123','$1$e8O/rKAf'),'Maildir','2010-02-16','01:22:00','dominio.com.br','Lucas
Testador');
```

E agora saia do cliente do MySQL:

```
mysql> quit
```

Rápida explicação das tabelas:
→ users - Tabela onde ficam os e-mails locais
→ aliases - Tabela que contém os alias dos e-mails
→ domain - Tabela que diz ao Sceo quais são os domínios locais
→ adomain - Tabela que contém os alias de domínios.

Vamos testar o sceo_mysql agora:

```
# ./sceo_mysql sceo_mysql.conf
```

Deve aparecer:

```
+OK SCEO_Mysql v 1.2.5 Conectado
```

Caso apareça “Não conectado” é por que você fez algo errado. Veja se o MySQL esta ligado ou se a senha foi configurada corretamente no sceo_mysql.conf.

Dentro do módulo digite:
```
user lucas@dominio.com.br
```
E ele responderá:
```
+OK
```

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

Digite:
```
quota lucas@dominio.com.br
```
```
+OK 100000000
```

Modulo e MySQL funcionando. Pressione CTRL + D para sair.

Dovecot

Sem rodeios. Vamos à instalação:

```
# cd /home/progs/
# tar xzf dovecot-1.2.14.tar.gz
# cd dovecot-1.2.14
# ./configure --with-mysql
# make
# make install
```

Crie um grupo e usuário para o Dovecot

```
# groupadd -g 106 dovecot
# useradd -s /bin/false -g dovecot -u 106 dovecot
```

Crie o arquivo /usr/local/etc/dovecot.conf:

```
# pico /usr/local/etc/dovecot.conf
```

Coloque o seguinte conteúdo dentro dele (alterando o hostname que esta ‘perdido’ ai no meio para o seu hostname correto):

```
base_dir = /var/run/dovecot/
protocols = imap pop3
listen = *
disable_plaintext_auth = no
ssl = no
mail_location = maildir:~
```

```
mail_uid = sceo
mail_gid = sceo
first_valid_uid = 105
last_valid_uid = 105

protocol pop3 {
 mail_plugins = quota
}
protocol imap {
 mail_plugins = quota imap_quota
}

protocol lda {
 postmaster_address = postmaster@dominio.com.br
 hostname = mx1.meudominio.com.br
 sendmail_path = /usr/local/sceo/sceo_mail
 mail_plugins = quota
}
auth_executable = /usr/local/libexec/dovecot/dovecot-auth
auth default {
 mechanisms = plain
 passdb sql {
  args =/usr/local/etc/dovecot-sql.conf
 }
 userdb sql {
  args = /usr/local/etc/dovecot-sql.conf
 }
 user = sceo
 socket listen {
  master {
   path = /var/run/dovecot/auth-master
   mode = 0600
   user = sceo
  }
 }
}
```

Salve e saia do arquivo.

Agora crie o arquivo /usr/local/etc/dovecot-sql.conf:

```
# pico /usr/local/etc/dovecot-sql.conf
```

Coloque o seguinte conteúdo:

```
driver = mysql

connect = host=localhost dbname=mail user=sceo password=minhasenha

default_pass_scheme = CRYPT

password_query = SELECT mail as name, domain, pass as password FROM users WHERE mail = '%u'

user_query = SELECT concat(home,maildir) AS home, uid, gid,concat('maildir') AS quota FROM users WHERE
mail = '%u'
```

ATENÇÃO:
As ultimas duas linhas fazem parte de uma linha só. Deixa-as em uma única linha

Salve e saia do arquivo.
Vamos criar o rc.dovecot para ligarmos e desligarmos o servidor.

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

Cole o seguinte conteúdo no arquivo:

```
#!/bin/sh
###############################################
# Script de controle do servidor Dovecot
# ./rc.dovecot start -> Ligar servidor
# ./rc.dovecot stop -> Parar servidor
# ./rc.dovecot restart -> Reiniciar servidor
#
#
#
ligar_dovecot() {
 echo "INICIANDO Dovecot..."
 /usr/local/sbin/dovecot
}
# Stop dovecot:
parar_dovecot() {
 echo "Parando Dovecot..."
 killall dovecot
}
case "$1" in
'start')
 ligar_dovecot
;;
'stop')
 parar_dovecot
;;
'restart')
 parar_dovecot
 sleep 1
 ligar_dovecot
;;
*)
echo "Os parametros aceitaveis sao: $0 start|stop|restart"
esac
#------- Fim do SCRIPT ---------
```

De permissão de execução:

```
# chmod 500 /etc/rc.d/rc.dovecot
```

Hora de testar os servidores. Mas antes é preciso criar o maildir daquele e-mail (lucas@dominio.com.br) que criamos.

```
# mkdir -p /home/mail/dominio.com.br/lucas/Maildir/new
# mkdir /home/mail/dominio.com.br/lucas/Maildir/cur
# mkdir /home/mail/dominio.com.br/lucas/Maildir/tmp
# touch /home/mail/dominio.com.br/lucas/Maildir/maildirsize
# chown -R sceo. /home/mail
# chmod -R 740 /home/mail
```

ATENÇÃO:

Acima eu usei os comandos chown e chmod desta forma porque é o primeiro e-mail criado. Lembre-se de que você precisa sempre ter tudo dentro do diretório **/home/mail** com permissão **740** e para o usuário '**sceo**', mas não cometa o erro de dar um '*chown -R*' e nem um '*chmod -R*' direto em /home/mail sempre que criar uma nova conta pois se você já tiver muitos domínios ou e-mails criados, o sistema operacional vai redefinir permissões de muitos e muitos arquivos e diretórios desnecessariamente.

Ligue os servidores agora:

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

É normal o Dovecot dar a mensagem abaixo antes de ser efetuado o primeiro login via pop/imap:

```
“If you have trouble with authentication failures,
enable auth_debug setting. See http://wiki.dovecot.org/WhyDoesItNotWork
This message goes away after the first successful login.”
```

De um ‘ps aux’ e na saída deve aparecer algo assim:

```
2727 ?        Ss     0:00 /usr/local/sceo/sceo
2728 ?        S      0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
2737 ?        S      0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
2739 ?        S      0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
2741 ?        S      0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
2743 ?        S      0:00 [SCEO_MODULE] /usr/local/sceo/mod/sceo_mysql.conf
2745 ?        S      0:00 /usr/local/sceo/sceo

2937 ?        Ss     0:00 /usr/local/sbin/dovecot
2938 ?        S      0:00 dovecot-auth
2939 ?        S      0:00 dovecot-auth -w
2941 ?        S      0:00 pop3-login
2942 ?        S      0:00 pop3-login
2943 ?        S      0:00 pop3-login
2944 ?        S      0:00 imap-login
2945 ?        S      0:00 imap-login
2946 ?        S      0:00 imap-login
```

Testando o Sceo:

```
# telnet localhost 25
220 mx1.dominio.com.br SMTP SCEO v0.30

ehlo localhost
250-mx1.dominio.com.br
250-SIZE 10485760
250-AUTH PLAIN LOGIN
250 8BITMIME

mail from: <lucas@dominio.com.br>
250 Remetente liberado

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

data
354 Envie o email e termine com <CRLF>.<CRLF>

From: Testando <lucas@dominio.com.br>
To: <lucas@dominio.com.br>

Subject: Teste

Ola mundo dos emails
Estou apenas fazendo um teste
.
250 Email aceito (ID: 1266282671_29640)

quit
```

```
221 Ate logo
```

Email entregue. Vamos testar agora o Dovecot:

```
# telnet localhost 110
+OK Dovecot ready.

user lucas@dominio.com.br
+OK

pass senha123
+OK Logged in.
```

Lembre-se de colocar a senha que usou na criação do email ‘lucas@dominio.com.br’ no INSERT de MySQL mais acima.

```
list
+OK 1 messages:
1 348
.

retr 1
+OK 348 octets
Return-Path: <lucas@dominio.com.br>
Received: from teste (localhost[127.0.0.1])
        by mx1.dominio.com.br SMTP SCEO v0.30b id 1266282671_mx129640
        for <lucas@dominio.com.br>; Tue, 16 Feb 2010 02:17:32 -0300
From: Testando <lucas@dominio.com.br>
To: <lucas@dominio.com.br>
Subject: Teste

Ola mundo dos emails
Estou apenas fazendo um teste

.

quit
+OK Logging out.
```

Nosso servidor já esta pronto para receber emails.
Percebi que muita gente tem dificuldades em entender o que é preciso para deixar o seu servidor MTA recebendo e-mails para um ou mais domínios. Por isso, vou dar uma rápida explicação com exemplos. Se você já entende como funciona o processo, pule a parte destacada abaixo.

Digamos que eu tenha o domínio meudominio.com.br e que o IP do meu servidor MTA seja 250.250.250.100. Então eu crio o nome mx1.meudominio.com.br (Em meu DNS)e aponto para o IP 250.250.250.100, depois eu crio um registro do tipo MX (Em meu DNS) com peso 10 que aponte para o endereço mx1.meudominio.com.br.
A lógica disto é: Sempre que um MTA for mandar um email para @meudominio.com.br, ele faz uma consulta em meu DNS perguntando quem é MX do domínio e terá como resposta o mx1.meudominio.com.br. Então aquele servidor conecta-se no 250.250.250.100, que é o nosso MTA, e entrega a mensagem.
É claro que em meu banco de dados MySQL eu preciso estar com o meudominio.com.br cadastrado na tabela ‘domain’ e as contas criadas na tabela ‘users’ e seus respectivos diretórios home criados.

Vamos agora a uma parte mais avançada do servidor. **Combate ao Spam**

SPF

O Sceo tem suporte nativo a SPF, basta usa-lo.
As respostas dos testes SPF se dão em números. Seguem os possíveis números:

```
Valor | 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)
```

Edite o arquivo regras_remetente:

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

Acrescente as seguintes linhas:

```
!Ip? "127.0.0.1" Spf_test!
Spf_resp? "4" Reply! "550 Remetente recusado. SPF Fail" Deny!
```

Obs: A exclamação (!) antes de uma condição inverte o resultado. Na regra acima (!Ip? "127....) ela será verdadeira se o IP conectado NÃO for o 127.0.0.1

Caso queria bloquear também servidores com SoftFail:

```
Spf_resp? "3" Reply! "550 Remetente recusado. SPF SoftFail" Deny!
```

Se quiser adicionar a resposta do SPF no header do email, acrescente também:

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

%y → Retorna o valor de resposta do teste SPF. Você pode usar esta variável nos outros arquivos de regras.

Salve e saia do arquivo.
SPF configurado.

Greylist

Alguns dizem que a Greylist vem caindo em desuso por atrasar mensagens válidas e por seu resultado final não ser tão eficiente. Eu discordo, a Greylist corta muito lixo eletrônico e se você souber quando usa-la, ela não atrasa mensagem de servidores bem configurados.
Para quem não sabe o que é a Greylist, aqui vai uma "rápida" explicação:

Existe um tipo de mensagem de erro no protocolo SMTP que indica um erro temporário, são os códigos que começam com 4 (Ex: 450, 451, etc..)
Se um servidor válido e bem configurado tentar entregar um email ao destinatário interno lucas@meudominio.com.br e obter a seguinte resposta de nosso MTA: "**450 Deu pau aqui e eu estou pegando fogo e NUNCA mais voce podera entregar este email a mim**", ele não dará a mínima para a mensagem em si pois o que importa é o código na frente, 450, que diz que ele deve tentar mais tarde e é exatamente isso que ele vai fazer, mas muitos programas de spam não tentam ou se tentam, as vezes eles mudam o endereço de remetente. É ai que a Greylist entra.

A Greylist verifica em um banco de dados se IP+remetente está liberado lá, se não estiver, a Greylist diz ao Sceo para responder com erro temporário (450 Tente mais tarde), se for um spammer é muito provável que ele não tente e se for um servidor válido e bem configurado ele vai tentar entregar o email novamente, então, nesta segunda tentativa o email será aceito e o IP+remetente será liberado para não sofrer mais este atraso.

Grande parte das reclamações da Greylist é o fato dela atrasar o recebimento dos emails, mas nós a usaremos só se o servidor remoto não passar pelo teste SPF, ou seja, todos os servidores bem configurados NÃO sofrerão o atraso da Greylist (Nao atrasando mensagens do Gmail, Hotmail, etc...) e outro diferencial é que não usaremos o endereço IP e sim o hostname que torna a Greylist muito mais eficiente.

Vamos a instalação:

```
# mkdir /usr/local/sceo/uteis
# cd /usr/local/sceo/uteis
# cp /home/progs/sceo_greylist.tar.bz2 .
# tar xjf sceo_greylist.tar.bz2
```

```
Criando o banco de dados:
# mysql
mysql> create database sceo_greylist;
mysql> grant all privileges on sceo_greylist.* to sceo@localhost identified by "minhasenha";
```

Lembre-se de colocar a mesma senha que usou para criação do banco de dados ‘mail’

```
mysql> flush privileges;
mysql> quit
# mysql sceo_greylist < sceo_greylist.sql
```

Agora você precisa colocar a senha do banco de dados no código fonte do sceo_greylist.

```
# pico sceo_greylist.c
```

Procure os defines ‘USUARIO’ e ‘SENHA’, altere seus valores colocando o usuário e senha do banco de dados sceo_greylist que criamos anteriormente.

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

Salve e saia do arquivo para compilarmos.

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

Se der erro na compilação é porque você não deve ter o Mysql-devel instalado ai em seu sistema operacional.
Vamos fazer o Sceo acionar o sceo_greylist caso o MTA conectado não tenha passado pelo teste SPF.

Edite o arquivo /usr/local/sceo/regras_destinatario:

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

Acrescente a seguinte linha de regra:

```
!Ip? "127.0.0.1" !Spf_resp? "1" !Auth? Internal_rcpt? Exec!
"/usr/local/sceo/uteis/sceo_greylist -i %s -f %f -t %r" Exec_resp? "1" Reply! "451 Tente
mais tarde" Deny!
```

OBS: As linhas passadas acima devem ficar em uma única linha dentro do arquivo.
Esta linha roda o programa sceo_greylist caso a resposta do SPF seja diferente de 1 (Pass), se a conexão NÃO for autenticada e se o destinatário for um email interno.

Caso queria processar Greylist para os MTAs que passarem pelo SPF também, basta tirar a segunda condição: *!Spf_resp? “1”*

Salve e saia do regras_destinatario. 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/local/sceo/uteis/sceo_greylist -clean*

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

Greylist instalada e configurada.

Clamav

Caso não queria instalar um antivírus em seu servidor de emails, basta pular esta sessão.

```
# cd /home/progs
# tar xzf clamav-0.95.3.tar.gz
# groupadd clamav
# useradd -g clamav -s /bin/false -c "Clam AntiVirus" clamav
# cd clamav-0.95.3
# ./configure --sysconfdir=/etc --libdir=/usr/lib
# make
# make install
```

Clamav instalado. Agora edite o arquivo /etc/clamd.conf:

```
# pico /etc/clamd.conf
```

Comente com um ‘#’ a linha ‘Example’ logo no início, deixando-a assim:

```
#Example
```

Salve e saia do arquivo. Faca o mesmo com o arquivo de configuração **/etc/freshclam.conf**. Este é o arquivo de configuração do programa de update do antivírus.
Ligue o Daemon do antivírus:

```
# /usr/local/sbin/clamd
```

Vamos colocar a linha acima no rc.local para o sistema operacional ligar o Clamav automaticamente.

```
# echo "/usr/local/sbin/clamd" >> /etc/rc.d/rc.local
```

Agora precisamos fazer o Sceo acionar o Clamav a cada email que chegar. Edite o arquivo de regras do Sceo que é processado assim que o email é entregue pelo MTA remoto. (/usr/local/sceo/regras_data):

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

Acrescente a seguinte linha: (**Obs: Os comandos abaixo devem ficar todos em uma única linha**)

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

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

Muitos “vírus” hoje em dia nos chegam por email em forma de link. Você deve estar cansado de ver as pessoas infectando seus computadores por clicar naqueles links que contém fotos de que estão sendo traídas, ou então, naquele link de recadastro no banco para sua conta não expirar.

Contra isto nós podemos implementar um recurso disponível para o Clamav e mantido pelo site www.malware.com.br.

Vou demonstrar abaixo uma técnica passada pelo Renato Martins.

Basta colocarmos o conteúdo do link ‘http://www.malware.com.br/cgi/submit?action=list_clamav’ em /usr/local/share/clamav/malware.db e reiniciarmos o clamd.

Vamos criar um script para esta tarefa e para aproveitar e atualizar o banco de dados de vírus:

```
# pico /etc/clamav_update.sh
```

Coloque o seguinte conteúdo:

```
#!/bin/sh

wget http://www.malware.com.br/cgi/submit?action=list_clamav -O  /usr/local/share/clamav/malware.db

chown clamav. /usr/local/share/clamav/malware.db

/usr/local/bin/freshclam --daemon-notify=/etc/clamd.conf
```

Salve e saia do arquivo.

```
# chmod 500 /etc/clamav_update.sh
# /etc/clamav_update.sh
```

Em nosso script, nós atualizamos o banco de dados de vírus e malware.

Agende o Cron para rodar este script uma vez por dia com a seguinte linha:

```
00 0 * * * /etc/clamav_update.sh
```

Spamassassin

Para quem não sabe, o Spamassassin é o filtro anti-spam. Ele vai filtrar boa parte do lixo eletrônico.
Antes de instalar o Spamassassin nós precisamos instalar os módulos que puxamos. São algumas ferramentas do Perl e que o anti-spam vai usar. No caso da minha distr. (Slackware 12.0 Full) eu não tive problemas para ‘compilar’ estes módulos. É possível que o Spamassassin peça por mais algum modulo necessário que a sua distr. não tenha. Neste caso você pode entrar no site http://search.cpan.org e procurar pelo módulo, puxar e instalar da mesma forma que instalo os outros aqui. (Caso não queira instala-lo, pule para a próxima parte Sceo_rquota)

Vamos ao trabalho:

```
# cd /home/progs
# tar xzf Digest-SHA1-2.12.tar.gz
# cd Digest-SHA1-2.12
# perl Makefile.PL
# make
# make install
```

```
# cd /home/progs
# tar xzf HTML-Parser-3.65.tar.gz
# cd HTML-Parser-3.65
# perl Makefile.PL
# make
# make install

# cd /home/progs
# tar xzf Net-DNS-0.66.tar.gz
# cd Net-DNS-0.66
# perl Makefile.PL
```
Aqui é capaz que ele te faça uma pergunta. CALMA !!! Sem pânico. É só dar ENTER.
```
# make
# make install

# cd /home/progs
# tar xzf NetAddr-IP-4.027.tar.gz
# cd NetAddr-IP-4.027
# perl Makefile.PL
# make
# make install

# cd /home/progs
# tar xzf IO-Zlib-1.10.tar.gz
# cd IO-Zlib-1.10
# perl Makefile.PL
# make
# make install
```

Esta terminando. Força ai ! Agora o Spamassassin:

```
# cd /home/progs
# tar xjf Mail-SpamAssassin-3.2.5.tar.bz2
# cd Mail-SpamAssassin-3.2.5/
# perl Makefile.PL
# make
# make install
```

Edite o arquivo de configuração do Spamassassin:

```
# pico /etc/mail/spamassassin/local.cf
```

Mude a opção *trusted_networks* para 127.0.0.0/8 e para sua rede local. Digamos que minha rede local seja 192.168.0.0/24, então ficaria assim:

```
trusted_networks 127.0.0.0/8 192.168.0.0/24
```

Acrescente a seguinte linha no final do arquivo:

```
skip_rbl_checks 1
```
Salve e saia do arquivo.
Spamassassin instalado e configurado, vamos agora criar um script para liga-lo.

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

Coloque o seguinte conteúdo:

```
#!/bin/sh

##############################################
# Script de controle do Spamassassin
#
spamd_start() {
  if [ -x /usr/bin/spamd ]; then
    echo "Starting spamd:   /usr/bin/spamd"
    /usr/bin/spamd -C /etc/mail/spamassassin -d -x -L --pidfile=/var/run/spamd.pid
  fi
}
spamd_stop() {
  pid=`cat /var/run/spamd.pid`
  kill $pid
}
case "$1" in
'start')
  spamd_start
  ;;
'stop')
  spamd_stop
  ;;
'restart')
  spamd_stop
  sleep 1
  spamd_start
  ;;
*)
  echo "Os parametros aceitaveis sao: $0 start|stop|restart"
esac
#---- Fim do Script ----
```

Salve e saia do arquivo e de permissão de execução para ele:

```
# chmod 500 /etc/rc.d/rc.spamd
```

Agora rode o script para ligar o Spamassassin.

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

Faça o rc.local chamá-lo quando o sistema iniciar:

```
# echo "/etc/rc.d/rc.spamd start" >> /etc/rc.d/rc.local
```

Caso queira testa-lo, chame-o assim: `spamc -c < email.eml`
Onde ‘email.eml’ é um arquivo texto que contém um email completo: header e corpo.

Nós temos duas opções para chamar o Spamassassin e faze-lo filtrar nossos emails:

Opção 1 - Chama-lo assim que o MTA remoto termina de entregar o email, com isso, a mensagem ainda pode ser recusada antes de ser aceita, mas a desvantagem é que se aquele email for para mais de um destinatario o bloqueio vai valer para todos e não há como separar a mensagem em pasta diferente (Pasta Spam).

Opção 2 - Chama-lo antes de entregar em cada caixa postal, através de um Shell Script acionado com a opção Dlocal do sceo.conf. Desvantagem é que servidor roda o Spamassassin mais vezes, uma para cada email interno, mas temos a vantagem de criar filtros individuais e se o email for considerado Lixo Eletrônico, ele vai parar na pasta Spam. Coisa que um Webmail ou Cliente MUA configurado via IMAP podem acessar sem problemas. Eu prefiro a opção 2. Vou demonstrar as duas aqui. Escolha uma delas:

OPÇÃO 1

Edite o arquivo /usr/local/sceo/regras_data

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

Acrescente no final do arquivo a seguinte regra: (**Obs: Todos os comandos abaixo devem ficar em uma única linha**)

```
!Ip? "127.0.0.1" !Auth? Exec! "/usr/bin/spamc -s 300000 -c < /var/spool/sceo/c%d"
Exec_resp? "1" Log! "/var/log/spamd.log: Spam -> %i  From:%f  To:%r" Reply! "500 Email
considerado Lixo Eletronico (Junk Mail)" Deny!
```

Caso a conexão não venha do localhost e não seja autenticada, o Sceo rodará o spamc para verificar a mensagem e recusar se for o caso e fazer um log (/var/log/spamd.log).

OPÇÃO 2

Edite o arquivo /usr/local/sceo/sceo.conf

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

Procure e altere a opção Dlocal que mudamos no inínio do tutorial para:

```
Dlocal "/usr/local/sceo/aplicafiltro %l"
```

Salve e saia do arquivo. Agora confirme a permissão de execução do arquivo /usr/local/sceo/aplicafiltro e edite-o:

```
# chmod 555 /usr/local/sceo/aplicafiltro
# pico /usr/local/sceo/aplicafiltro
```

Deixe-o desta forma:

```
#!/bin/sh

/usr/bin/spamc -s 300000 -c < $2
RESP=$?

if [ $RESP == 0 ]; then
 # --- Caso O Spamassassin Tenha retornado ZERO (Nao e' Spam) ---
 /usr/local/libexec/dovecot/deliver -d $1 < $2
else
 # --- Caso O Spamassassin Tenha retornado 1 (É' Spam) Vamos separar na pasta Spam ---
 /usr/local/libexec/dovecot/deliver -m Spam -d $1 < $2
fi
# --- Fim do Script ---
```

Salve e saia do arquivo. Esta feito. Se o Spamassassin considerar a mensagem como Lixo, o deliver vai entregar a mensagem na pasta Spam dentro da caixa postal do cliente.

Sceo_rquota

O Sceo_rquota é um programa externo que permite a você limitar a quantidade de mensagens que uma conta de email pode enviar por dia/semana/mês. O controle pode ser feito por domínio, remetente, usuário autenticado, IP, etc...

Isto protege o seu servidor impedindo que algum cliente engraçadinho o use para enviar spam.

É muito simples instalar e configurar. Vamos joga-lo no diretório /usr/local/sceo/uteis criado anteriormente.

```
# cd /usr/local/sceo/uteis
# cp /home/progs/sceo_rquota.tar.bz2 .
# tar xjf 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";
mysql> flush privileges;
mysql> quit
```

ATENÇÃO:

Lembre-se de colocar a mesma senha que usou para a criação do banco de dados 'mail' lá no começo. Se você mudar a senha aqui, o sceo_module usado pelo Sceo vai parar de funcionar e ai eu vou receber um email seu com o assunto: "Mas não funciona..."

Crie as tabelas usando o arquivo sceo_rquota.sql que acompanha o pacote:

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

Edite o sceo_rquota.c:

```
# pico sceo_rquota.c
```

Procure e altere as defines USER e PASS, colocando o usuário e senha do banco de dados sceo_rquota criado acima:

```
#define USER "sceo"
#define PASS "minhasenha"
```

Salve e saia do arquivo e já podemos compilar o programa:

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

Se der alguma mensagem de erro é porque você não tem o Mysql-devel instalado.

Abra o arquivo de regras que o Sceo executa sempre que o remetente é especificado: /usr/local/sceo/regras_remetente

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

Acrescente a seguinte linha no inicio do arquivo: (**Obs: Todos os comandos devem ficar em uma única linha**)

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

Salve e saia do arquivo

Explicação da regra:

Primeiro ele verifica se a atual conexão é autenticada e depois se o remetente não é nulo e então executa o programa externo /usr/local/sceo/uteis/sceo_rquota com a opção -c e -u <email do remetente>. Este programa faz uma consulta no BD para saber se remetente já estourou o limite de envio. Depois a função Exec_resp? testa o valor retornado pelo ultimo programa executado e se for 1, o servidor responde “500 Limite de envio ....” e nega o remetente impedindo assim o envio do email. O sceo_rquota retorna 1 se o email estourou o limite e ZERO caso contrário.

Mas ainda falta uma configuração. Precisamos fazer o sceo_rquota atualizar no BD a quantidade de emails enviada pelo remetente. Abra o arquivo de regras que o Sceo roda assim que o cliente termina de enviar o email: /usr/local/sceo/regras_data

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

Coloque a seguinte regra no final do arquivo:

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

Salve e saia do arquivo.

Explicação da regra:

Primeiro ele verifica se a atual conexão é autenticada e depois se o remetente não é nulo e então executa o sceo_rquota com a opção -a <numero de destinatários do email> -u <email do remetente> -d 1000.
O valor 1000 é a quota padrão a ser criada caso o remetente ainda não tenha um registro no BD. Você pode alterar a quota de um email já registrado no banco de dados a hora que quiser.

Agende seu cron para rodar o sceo_rquota de forma a zerar o número de destinatários enviados de cada remetente ou dominio, etc.
Você pode agendar o cron uma vez ao dia, semana, mês ou da forma que quiser. Escola uma das formas abaixo.

Exemplos de configurações no Cron.
Por dia:
```
00 0 * * * /usr/local/sceo/uteis/sceo_rquota -z
```

Por semana:
```
00 0 * * 0 /usr/local/sceo/uteis/sceo_rquota -z
```

Por mês:
```
00 0 1 * * /usr/local/sceo/uteis/sceo_rquota -z
```

Servidor finalizado. Não se esqueça de chamar o Sceo e o Dovecot quando o sistema ligar:

```
# echo "/etc/rc.d/rc.sceo start" >> /etc/rc.d/rc.local
# echo "/etc/rc.d/rc.dovecot start" >> /etc/rc.d/rc.local
```

O envio de e-mails não ficará instantâneo porque o Sceo esta rodando o antispam e antivírus. Se você quiser, pode incluir condições na frente das regras para não rodar antivírus ou antispam para sua rede local (!Ip? “192.168.0.0/24”) ou para clientes autenticados (!Auth?).
Outra coisa que pode fazer o envio de e-mails não ficar instantâneo é o fato do servidor demorar para resolver o nome do IP de uma rede local. Se a sua rede é muito grande e você não quer ficar colocando os nomes dos IPs em /etc/hosts, você pode mandar o Sceo não resolver o endereço reverso para IPs da rede local, assim:

Abra o /usr/local/sceo/sceo.conf

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

Altere a opção Resolve_all deixando-a desativada, assim:

```
Resolve_all 0
```

Salve e saia do arquivo.
Edite o arquivo /usr/local/sceo/regras_conexao

```
# pico /usr/local/sceo/regras_conexao
```

Digamos que a minha rede local seja 192.168.0.0/24. Acrescente a seguinte regra logo no inicio do arquivo:

```
!Ip? "192.168.0.0/24" Resolve!
```

Salve e feche o arquivo.
Reinicie o Sceo:

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

Teste agora o tempo de envio de seu cliente de e-mail ao seu servidor SMTP.

Técnicas Diversas

Esta eu acho que esta é a sessão mais interessante. Aqui você aprenderá um pouco de como configurar o Sceo para se defender de lixo eletrônico, ataques e etc.
Mas lembre-se de que o seu servidor **JÁ ESTÁ** configurado e pronto para funcionar e você não precisa colocar as regras descritas aqui.

➔ Anti-DoS

Este tipo de ataque consiste em estabelecer muitas conexões em determinado serviço de um servidor fazendo-o negar novas conexões (Denial of Service).

Podemos tomar algumas medidas para dificultar ao máximo esse tipo de ataque.
No Linux, podemos ligar o tcp_syncookies para fazer o kernel só alocar recursos para atender a conexão se receber de volta o pacote SYN / ACK. (Para mais detalhes, procure em seu buscador por ‘Syn Flood DoS’)

Execute o seguinte comando:

```
# echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Esta feito. Não é preciso reiniciar nada. Se quiser você pode colocar o comando acima no seu rc.local:

```
# echo "echo 1 > /proc/sys/net/ipv4/tcp_syncookies" >> /etc/rc.d/rc.local
```

A opção passada acima já te da uma boa segurança, mas pode haver casos de tentativas de DoS onde as conexões sejam realmente estabelecidas e não apenas um Syn Flood e nestes casos o tcp_syncookies se torna inútil.

Para estes casos nós podemos fazer o Sceo bloquear o IP do atacante no Firewall do servidor automaticamente. Crie então uma Chain chamada BLACKLIST em seu Firewall, pois é lá que incluiremos os IPs bloqueados:

```
# iptables -N BLACKLIST
# iptables -I INPUT -d ! 127.0.0.1 -j BLACKLIST
```

Obs: Você deve colocar as linhas acima TAMBÉM em seu script de Firewall (geralmente /etc/rc.d/rc.firewall) para ele criar automaticamente esta Chain e chamá-la quando o servidor estiver ligando. Como eu não sei como é o seu Firewall eu passei regras meio genéricas. Se você entende de iptables e sacou a sentido da coisa, configure da forma correta em seu Firewall...

Nas regras de iptables acima eu primeiro criei uma Chain chamada BLACKLIST e na linha de baixo eu mandei inserir uma regra logo no inicio mandando o kernel processar a Chain BLACKLIST caso o IP de destino do pacote NÃO seja o localhost.

Agora vamos às configurações no Sceo.

Abra o arquivo /usr/local/sceo/sceo.conf:

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

Procure e altere as opções Limit_buf, Limit_cnx e Limit_time conforme os valores abaixo:

```
Limit_buf 100
Limit_cnx 20
Limit_time 2
```

Explicação:

O sistema de limites é ligado quando definimos um valor diferente de zero para Limit_buf, ela diz ao Sceo que ele deve analisar as ultimas 100 conexões a procura de IPs que estabeleceram no mínimo 20 conexões em um período inferior a 2 segundos entre uma conexão e outra. Estes valores variam bastante entre servidores com grande, médio ou baixo fluxo de emails. Sinta-se a vontade em alterar os valores conforme a necessidade. De início, use estes valores e fique de olho no Log que vamos "montar" para monitorarmos e depois altere os valores se quiser.

Certifique-se de que a opção Limit_rules esteja com o parâmetro `"/usr/local/sceo/regras_limite".` Este é o arquivo de regras que o Sceo vai executar caso um possível DoS tenha sido detectado.
Salve e saia do sceo.conf. Agora abra o arquivo /usr/local/sceo/regras_limite:

```
# pico /usr/local/sceo/regras_limite
```

Neste arquivo nós mandaremos o Sceo incluir uma regra no Firewall para bloquear o IP em questão, mas devemos tomar muito cuidado pois isto pode bloquear outros servidores que fazem parte de nossa rede ou de confiança e que geralmente entregam grandes quantidades de emails. No exemplo que darei abaixo eu vou mandar ele aceitar e parar de processar o regras_limite caso o IP seja de minha rede local ou de alguns outros servidores cujo IP eu escolhi aleatoriamente para o exemplo:

IPs locais: 127.0.0.1, 10.10.10.0/24
IPs de outros servidores: 240.240.240.2, 240.240.240.3 e 70.70.70.70

Coloque no arquivo o seguinte conteúdo:

```
Ip? "127.0.0.1" Accept!
Ip? "10.10.10.0/24" Accept!
Ip? "240.240.240.1" Accept!
Ip? "240.240.240.2" Accept!
Ip? "240.240.240.3" Accept!
Ip? "70.70.70.70" Accept!
Log! "/var/log/sceo/sceo_limit.log:%i Bloqueado"
Exec! "iptables -I BLACKLIST -s %i -j DROP"
```

O Sceo executa este arquivo de regras ainda com permissão de root, para depois descer para a permissão apontada pela opção 'User' do arquivo sceo.conf

Agora nós temos que agendar o Cron para zerar a Chain BLACKLIST 1 vez por dia para estes IPs não ficarem bloqueados para sempre ou até o próximo reboot do sistema. Acredite, isto é necessário para o Firewall não ficar lento com o tempo, porque um IP não deve ser punido para sempre e tudo isto foi criado apenas para impedir um tipo de ataque que acontece por um curto período do dia.

Acrescente a seguinte regra em seu Cron:

```
00 0 * * * iptables -F BLACKLIST
```

Reinicie o Sceo:

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

Esta feito. Se por acaso você quiser que o Sceo te mande um email a cada IP bloqueado, acrescente a seguinte linha de regra no final do regras_limite (**Obs: As regras abaixo devem ficar em uma única linha e não esqueça de colocar o seu endereço de email no lugar**)

```
Exec! "/usr/local/sceo/sceo_mail -to eu@meudominio.com.br -f eu@meudominio.com.br -su
DoS_de_%i -sf /usr/local/sceo/aviso_DOS.eml"
```

Agora crie o arquivo `/usr/local/sceo/aviso_DOS.eml` com o seguinte conteúdo:

```
Uma tentativa de DoS foi detectada no servidor SMTP
e seu acesso foi bloqueado pelo Sceo e seu IP cortado
no Firewall

Verifique o Log: /var/log/sceo/sceo_limit.log
para maiores detalhes
```

Salve e saia do arquivo. Esta Feito !

➔ **Helo 'Eu sou você'**

É engraçado quando você esta olhando o Log de seu servidor e vê que IPs externos conectam-se e no comando 'helo/ehlo' dizem ser você.

Digamos que o IP de seu servidor seja 240.240.240.1 e o FQDN seja mx1.meuservidor.net. Então você deve cortar qualquer conexão que mande um 'helo 240.240.240.1' ou 'helo [240.240.240.1]'. Edite o arquivo /usr/local/sceo/regras_remetente:

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

E acrescente o seguinte conteúdo no final do arquivo:

```
!Ip? "127.0.0.1" !Ip? "240.240.240.1" Helo? "240.240.240.1" Reply! "550 Cai fora SPAMMER" Close!
!Ip? "127.0.0.1" !Ip? "240.240.240.1" Helo? "[240.240.240.1]" Reply! "550 Cai fora SPAMMER" Close!
!Ip? "127.0.0.1" !Ip? "240.240.240.1" Helo? "mx1.meuservidor.net" Reply! "550 Cai fora SPAMMER" Close!
```

Explicação:
Caso o IP conectado NÃO seja o 127.0.0.1 e 240.240.240.1 e no Helo ele mandar 240.240.240.1 ou [240.240.240.1] ou mx1.meuservidor.net, o Sceo responderá '550 Cai fora SPAMMER' e fechará a conexão.

Esta regra você pode colocar sem dó. Só spammer cai nela

➔ **Blacklist via RBLs mundiais**
Abra o arquivo regras_conexao:

```
# pico /usr/local/sceo/regras_conexao
```

Acrescente as regras abaixo no regras_conexao para fazer o Sceo fazer consultas em RBLs

```
Ip? "127.0.0.1" Stop!

Rbl_test! "bl.spamcop.net zen.spamhaus.org b.barracudacentral.org"

!Rbl_resp? "0" Log! "/var/log/sceo/blacklist.log: Bloqueado %i %f %r %R->%T" Reply! "550 IP
em Blacklist RBL %T" Close!
```

Eu separei por uma linha em branco as regras acima para você saber quais 'comandos' devem ficar em uma única linha.

A primeira linha de regra: `Ip? "127.0.0.1" Stop!` faz o Sceo PARAR de processar o arquivo regras_conexao caso o IP conectado seja o 127.0.0.1. Você deve fazer isso para o IP local de Internet do servidor e também para o seu bloco de rede local.

Embora você tenha feito a checagem RBL no regras_conexão, você não precisa necessariamente bloquear a conexão aqui. Tenho caso de clientes que QUEREM receber SPAMs e eu não posso rodar Blacklist para ele. Como resolver isto, já que o endereço do remetente só é especificado bem depois ?

SIMPLES ! Digamos que o domínio do cliente que ama Spams seja 'amospam.com.br'. Então basta remover a linha (`!Rbl_resp? "0" ...`) ai do regras_conexao e recoloca-la no regras_destinatario da seguinte forma:

```
!Find_in_to? "@amospam.com.br" !Rbl_resp? "0" Log! "/var/log/sceo/blacklist.log:
BLACKLIST_RBL %i %f %r %R->%T" Reply! " 505 IP em Blacklist RBL %T" Close!
```

Explicação:
Caso NÃO seja um email com destino ao domínio 'amospam.com.br' o Sceo vai verificar qual foi a resposta das RBLs testadas durante o regras_conexao e cortar ou não a conexão.

É possível criar um arquivo texto contendo uma lista de domínios que amam Spam e fazer o Sceo consulta-lo. Mas este tutorial já esta ficando muito extenso e tenho mais coisas para passar. Você pode acessar o site oficial do Sceo (http://www.sceo.com.br) e aprender tudo isso e muito mais no fórum de lá.

➔ **Blacklist local**
Você pode montar a sua Blacklist local, contendo aqueles nomes de servidores, IPs ou remetentes que insistem em entregar aquelas incríveis promoções que adoramos receber .

Vou montar aqui uma blacklist contendo nomes de servidores.
Crie um arquivo texto contendo o nome ou parte do nome dos servidores a bloquear.

```
# pico /usr/local/sceo/blacklist_srvs.txt
```

Vou colocar alguns nomes como exemplo dentro do arquivo:

```
.hospedagemdespammers.ws
mx1.spamlandia.net
supermailmarketing.com.br
.spammersunidos.
```

Salve e saia do arquivo.
Edite o regras_conexao:

```
# pico /usr/local/sceo/regras_conexao
```

Acrescente a seguinte regra no fim do arquivo, em apenas uma única linha:

```
Find_strfile? "%e:/usr/local/sceo/blacklist_srvs.txt" Reply! "500 Host spammer. Blacklist"
Close!
```

➔ **Limitar o número de destinatários por email de um remetente nulo:**

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

Acrescente o seguinte conteúdo logo no começo do arquivo:

```
From? "" Itest? "%n>1" Reply! "550 Muitos rcpts para um remetente nulo" Deny!
```

➔ **Só aceite remetentes internos se eles se autenticarem**
Se o remetente é um cliente entregando mensagem em seu servidor, por que não autenticar-se ?
Muitos Spams nos chegam com o endereço de remetente igual ao do destinatário. Aceitando somente remetentes locais autenticados você impede este tipo de Spam.

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

Acrescente o seguinte conteúdo logo no começo do arquivo:

```
!Ip? "127.0.0.1" Internal_from? !Auth? Reply! "550 Voce precisa autenticar-se" Deny!
```

➔ **Inventando um SPF para o Yahoo**
Muitos Spams nos chegam com endereço de remetente sendo @yahoo.com. Para piorar a situação o Yahoo NÃO descreve SPF em seu DNS para sabermos quem é que pode ou não enviar emails @yahoo.com.

Vamos então “inventar” um SPF para ele, baseado no FQDN que todos os MTAs do Yahoo tem usado para entregar emails. Isto não da 100% de certeza de que o email seja do Yahoo pois este nome pode ser forjado, mas dificulta bastante a prática de Spams que usam @yahoo como remetente.

Abra o arquivo /usr/local/sceo/regras_remetente

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

Acrescente o seguinte conteúdo no inicio e em uma única linha:

```
Find_in_from? "@yahoo.com" !Find_in_reverse? "yahoo.com" Reply! "550 Voce nao parece
pertencer a rede Yahoo. Desculpe" Deny!
```

Salve e saia do arquivo.
Não é preciso reiniciar o Sceo. Você só precisa reinicia-lo quando alterar o sceo.conf.

➔ Recusando Helo diferente do FQDN

O Correto é informar o FQDN durante o comando ‘helo/ehlo’ do SMTP. Nós podemos recusar servidores que cometam esta transgressão. É uma medida meio radical mas cada caso é um caso.

Edite o arquivo regras_remetente

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

Coloque a seguinte regra logo no início e em uma única linha:

```
!Ip? "127.0.0.1" !Ip? "240.240.240.1" !Cmp_reverse "%h" Reply! "500 CMD Helo recusado" Deny!
```

Lembre-se de colocar no lugar de 240.240.240.1 o IP do seu servidor ou então da sua Rede.

➔ Fazendo backup de todas as mensagens que passam pelo servidor.

Vi que existe muita gente preocupada em manter um backup de todas as mensagens que passam pelo servidor e a técnica usada é copiar cada mensagem recebida por cada destinatario em outra caixa postal ou de tempos em tempos copiar o maildir inteiro do servidor ou até fazer uma copia do email em outro local com complicados scripts que rodam no MUA (procmail, maildrop, deliver, etc...). Mas se o seu servidor recebe uma mensagem de 2 Mb para 30 contas de emails locais, no final você estará fazendo backup de 60 Mb. Isto é totalmente desnecessário. Você pode fazer backup apenas de 1 mensagem de 2Mb e se precisar usa-la você sabe onde encontrar.

Vamos fazer um Shell Script para o Backup:
Crie e edite o arquivo /usr/local/sceo/uteis/backup.sh

```
# pico /usr/local/sceo/uteis/backup.sh
```

Acrescente o seguinte conteúdo:

```
#!/bin/sh

ANO=`date +%Y`
MES=`date +%b`
DIA=`date +%d`

mkdir -p /mnt/backup/mail/$ANO/$MES/$DIA > /dev/null
cp /var/spool/sceo/*$1 /mnt/backup/mail/$ANO/$MES/$DIA
```

```
# --- Fim do Script ---
```

Salve e feche o arquivo.

De permissão de execução:

```
# chmod 555 /usr/local/sceo/uteis/backup.sh
```

Crie o diretorio de backup com permissão 777:

```
# mkdir /mnt/backup
# chmod 777 /mnt/backup
```

Abra o arquivo regras_data que é processado cada vez que um email inteiro é entregue:

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

Acrescente a seguinte regra no final:

```
Exec! "/usr/local/sceo/uteis/backup.sh %d"
```

Salve e feche o arquivo.

Pronto ! O Sceo vai rodar o backup.sh cada vez que um email chegar e o script se encarrega de criar automaticamente a estrutura de backup para a data atual e armazena o arquivo de email e sua lista de destinatários lá.
Nos arquivos de fila do Sceo, os que começam o ‘c’ indicam o conteúdo do email, ou seja, é o email completo contendo header e corpo e os que começam com ‘l’ armazenam os destinatários daquele email.

➔ **Usando HITS para pontuar negativamente o host.**
Todas as ténicas de combate anti-spam passadas acima cortam o host conectado em qualquer transgressão, mas pode ser que ao invés de cortar logo de cara o host apenas por cometer um erro, você queira corta-lo se cometer dois ou mais erros.
Para isto nós usamos a ação Add_hits! para adicionar ou tirar pontos, e depois testamos a variável %H para cortar ou não a conexão.

Vou repassar algumas regras descritas em sessão anteriores de forma que usem o Hits.

Coloque as seguintes regras no regras_conexao:

```
Rbl_test! "bl.spamcop.net zen.spamhaus.org b.barracudacentral.org"
!Rbl_resp? "0" Add_hits! "1" Add_hits! "%R"
```

Se o Spammer for encontrado em alguma RBL, ele vai ganhar 2 ou mais pontos negativos. A variavel %R retorna o número de RBLs em que ele foi encontrado, ou seja, eu adicionei 1 ponto negativo e logo em seguida mandei adicionar como ponto negativo o número de RBLs em que ele foi encontrado.

Salve e saia do arquivo.

Agora coloque as seguintes regras no final de regras_remetente (Estou levando em conta que você já esta com a regra de fazer o teste SPF --> Spf_test!, que foi configurada anteriormente no meio deste tutorial):

```
!Spf_resp? "1" Add_hits! "1"
!Ip? "127.0.0.1" !Ip? "240.240.240.1" Helo? "240.240.240.1" Add_hits! "2"
!Ip? "127.0.0.1" !Ip? "240.240.240.1" !Cmp_reverse "%h" Add_hits! "1"
Itest? "%H>2" Reply! "550 Voce acumulou muitos pontos negativos" Deny!
```

Salve e saia do arquivo.
Se por acaso o host conectado ganhar 3 pontos ou mais ele será negado para entregar emails
A Condição Itest? verifica se o valor da variavel %H é maior que 2, se for, o Sceo nega o host conectado.

É isso ai. Seu servidor de emails esta terminado.

Espero ter ajudado e que tenha gostado do tutorial e que goste do projeto MTA Sceo.

Gostaria de receber dicas de melhorias, sinta-se a vontade para me mandar suas idéias, mudanças e correções.

Obrigado a todos.

Lucas Priori
lpriori@hospedaria.com.br

Site dos projetos e referências:

MTA Sceo (http://www.sceo.com.br)
Dovecot (http://www.dovecot.org)
Clamav (http://www.clamav.net)
Spamassassin (http://spamassassin.apache.org)