SQUID 3.4.8 + SSL + FIREWALL + DNS + DHCP NO DEBIAN JESSIE (V.8)

Autor: Riccelli Reis de Oliveira <r2oweb at gmail.com>
Data: 07/03/2016

ORIENTAÇÕES INICIAIS

Sim! Este é mais um artigo que mostra como bloquear/gerenciar acesso a sites "seguros", com protocolo SSL, onde a URL começa com HTTPS. A diferença é que os procedimentos foram testados por semanas e o servidor está em perfeito funcionamento. Outra vantagem é que o artigo está atualizado, os últimos testes no servidor foram realizados dia 25 de fevereiro de 2016. Depois de meses pesquisando e testando, enfim fizemos o Squid realmente trabalhar bem com SSL.

No final do artigo tem um link para uma página que tem só os comandos, sem as explicações.

Hardware Mínimo Recomendado: CPU Intel Core i3, RAM 4GB DDR3, HD 500GB Sata2, Placa de Rede OffBoard, sem teclado, sem mouse, sem monitor, dois cabos UTP conectados em Switch de 100Mpbs, link para internet com

fibra optica.

Software Utilizado: Debian 8 (Jessie) atualizado (apt-get upgrade) até 25/02/2016, firewall Iptables, Samba, ISC-DHCP-Server, DNS Bind9, Script CBQInit, Squid 3.4.8, IfTop, Cacti, Net Monitor, GLPI com OCS Inventory e Webmin.

Dicas:

1) Prepare um computador exclusivo para ser o servidor. Deixe-o protegido de usuários/curiosos. Se preocupe bastante com a energia, o ideal é um NoBreak com bateria externa e aterramento. Se tiver condições, adquira uma máquina nova, de configuração boa, de preferência um servidor Dell ou HP.

2) Se possível, instale um sistema operacional novo, não instale nem configure mais nada, apenas o que vamos indicar. Depois da instalação e configuração do Squid, também indicamos configurações para a rede, Firewall, DNS e DHCP. Em outra oportunidade publicaremos outro artigo sobre controle de banda e monitoramento.

3) Mantenha o CD/DVD de instalação no leitor, ou copie todo o conteúdo para um diretório. Mais a frente explicamos.

4) Preste bastante atenção pois todos os comandos estão nas linhas que começam com root@servidor#. Você pode copiar tudo depois de # . Outras linhas que começam com # são apenas comentários. Se algo der errado, releia e refaça.

5) Parece muito óbvio para os experientes, mas não esqueça de teclar Enter depois de cada comando.

6) Para editar os arquivos de configuração usamos o VIM. Você pode usar o Nano ou qualquer outro de sua preferência. Observe que cada editor tem seu modo de salvar o arquivo. No VIM, para iniciar a edição digite i, altere o que precisa, depois digite ESC, depois :wq!Enter (dois pontos, wq, exclamação, Enter)

7) Fique atento(a)! Se aparecer alguma mensagem de erro, leia novamente, copie as linhas com mais cuidado e repita o processo. De preferência escreva nos comentários que ajudamos a resolver, isso pode ajudar outras pessoas com o mesmo problema.

8) Leia atentamente cada passo.

INSTALAÇÕES

Passo 1) Configurar os repositórios. Se preferir, copie o conteúdo da ISO do Debian para o HD e especifique no sources.list. Fizemos isso, criando o diretório repo, em /var. O local onde o sistema monta o DVD pode mudar um pouco de acordo com a Distribuição que você usa, então substituímos esta localização por reticências (...). Verifique onde seu servidor montou o DVD e substitua os 3 pontos abaixo pela sua localização.

```
# mkdir /var/repo
# cp /.../* /var/repo
# vim /etc/apt/sources.list
```

```
deb file:/var/repo/ jessie contrib main
#Oficial
deb http://ftp.debian.org/debian/ jessie main non-free
contrib
deb-src http://ftp.debian.org/debian/ jessie main
non-free contrib
deb http://security.debian.org/ jessie/updates main
contrib non-free
deb-src http://security.debian.org/ jessie/updates main
contrib non-free
# jessie-updates, previously known as 'volatile'
deb http://ftp.debian.org/debian/ jessie-updates main
```

```
contrib non-free
deb-src http://ftp.debian.org/debian/ jessie-updates
main contrib non-free
#Personal
deb http://ftp.br.debian.org/debian/ jessie main contrib
non-free
deb-src http://ftp.br.debian.org/debian/ jessie main
contrib non-free
deb http://ftp.debian.org/debian/ jessie-proposed-
updates main contrib non-free
deb http://ftp.br.debian.org/debian/ jessie-updates main
contrib non-free
deb-src http://ftp.br.debian.org/debian/ jessie-updates
main contrib non-free
#Backports
deb http://http.debian.net/debian jessie-backports main
```

Passo 2) Atualizar o sistema. Mesmo que apareça algum alerta, continue.

```
# apt-get update
# apt-get upgrade
```

Passo 3) Entrar no diretório indicado para salvar os instaladores.

```
# cd /usr/src
```

Passo 4) Instalar as dependências. Nem tudo é necessário, mas nosso servidor tem Firewall, DNS, DHCP, Proxy, Controle de Banda, entre outros recursos. Aconselhamos instalar tudo para evitar erros mais a frente, mesmo que não vá usar estas instalações não irão atrapalhar, mas se você souber exatamente o que não vai usar retire os nomes apropriados das linhas abaixo. Dependendo da sua conexão com a internet, pode levar de 10 minutos a 2 horas para instalar tudo.

```
# apt-get install apache2 php5 bind9
binutils build-essential fakeroot
devscripts libssl-dev gawk wget udev
libaal-dev gcc-multilib automake gzip
dpatch gdebi-core isc-dhcp-server libc6
libc6-dev libcap-dev libcrypto++-dev
libecap2-dev libncurses5-dev libpam-
modules libpam-chroot libpam0g
libpam0g-dev libdb-dev cdbs
libsasl2-dev libcppunit-dev libkrb5-dev
comerr-dev libcap2-dev libexpat1-dev
libxml2-dev autotools-dev libltdl-dev
pkg-config libnetfilter-conntrack-dev
logrotate nettle-dev libaal-dev
libncurses5-dev dpatch automake
```

```
gdebi-core libpam-chroot squid-langpack
samba smbclient vim ssl-cert
```

Passo 5) Instalar apenas as dependências do Squid e do OpenSSL.

```
# apt-get build-dep squid3 openssl
```

Passo 6) Obter o código fonte do Squid para depois compilar com recursos que não seriam instalados com o apt-get install ou aptitude install.

```
# apt-get source squid3
```

Passo 7) Entrar no diretório criado automaticamente após o comando anterior:

```
# cd squid3-3.4.8
```

Passo 8) Editar o arquivo rules para especificar os recursos adicionais na instalação. A linha 46 termina com --with-default-user=proxy. Coloque uma barra invertida (\) no final dela e adicione mais 3 (três) linhas, igual abaixo. Observe que na última não tem a barra.

```
# vim debian/rules
```

```
--with-default-user=proxy \
--enable-ssl \
--enable-ssl-crtd \
--with-openssl
```

Passo 9) Editar o arquivo changelog e adicionar informações sobre a versão da nova compilação, seus dados, email, etc. Dica: Copiar as 9 primeiras linhas do arquivo original, colar no início do arquivo e alterar apenas os textos, como abaixo. Não apague os símbolos, como arroba (@), traço(-), etc. Em alguns testes, quando digitamos novas linhas, talvez por causa do sistema de caracteres, o compilador apresentou erros relativos a esse arquivo. Observe que no lugar de "Seu Nome" e "seuemail", você vai colocar as suas informações pessoais, nada comprometedor então é aconselhável preencher corretamente. Coloque também a data e hora corretas.

```
# vim debian/changelog
```

```
squid3 (3.4.8) jessie; urgency=high

  [ Seu Nome <seuemail> ]
  * debian/rules
    - Added --enable-ssl --enable-ssl-crtd (Closes:
```

```
#800000)
```

-- Seu Nome <seuemail> Wed, 22 Jul 2015 18:36:08 +0200 <<< Mudar para a data e hora atual, mas não deixe este texto explicativo no seu arquivo.

Passo 10) Preparar/Configurar primeiro. Geralmente isso só é feito quando você vai compilar pelos meios tradicionais, que seriam os comandos make e make install. Vamos usar outro método que diminui incompatibilidades. No próximo passo explicamos melhor. Digite a linha abaixo e aguarde uns 2 minutos.

```
# ./configure
```

Passo 11) Compilar e já criar os instaladores no formato padrão do Debian (.deb). Isso garante que tudo vai ser colocado nos locais padrão, como se fosse instalado com apt-get. Leva de 10 a 15 minutos. No final, será fácil confirmar se deu certo, no máximo será exibida uma mensagem de alerta (Warning).

Se algo der errado, levará menos tempo e vai aparecer uma mensagem de erro (Error), fácil entender mesmo que você não saiba nada de inglês. Neste caso repita todos os procedimentos desde o começo do artigo. Primeiro ele vai repetir as configurações do passo anterior (./configure), mas é só para confirmar que tudo vai dar certo. Quando

terminar, deve aparecer isso: N: 1 tag overridden (1 warning) e Finished running lintian.

```
# debuild -us -uc -b
```

Passo 12) Voltar para o diretório anterior/acima, pois foi onde os instaladores foram salvos.

```
# cd ..
```

Passo 13) Instalar o Squid e outros pacotes necessários que foram criados. São seis e deve levar uns 2 minutos.

```
# dpkg -i *.deb
```

Passo 14) Travar o Squid para que não atualize. Se você rodar o comando apt-get upgrade depois que instalar o Squid, com certeza o sistema vai dizer que é necessário atualizá-lo, só que com isso ele perde a capacidade para trabalhar com SSL. Os pacotes squid-cgi, squid-purge e squidclient serão atualizados e não causam problemas para nossa instalação. Se for necessário, daqui a um ano ou dois, se sair alguma versão com grandes mudanças, pensamos em recompilar o Squid e adaptar a configuração atual para a versão nova. Por enquanto é melhor deixar assim.

```
# apt-mark hold squid3
```

CONFIGURAÇÃO DO SQUID - PARTE 1

Antes do *Squid* trabalhar, é necessário que as conexões sejam direcionadas pelo Firewall, para o Squid. Mais a frente deixamos um exemplo muito bom de Script de Firewall. Se preferir, apenas utilize as duas linhas abaixo para direcionar os clientes ao Squid. Lembre que dessa forma, se o sistema for reiniciado ou acontecer algum problema com a rede, o redirecionamento para o Squid não acontece. As linhas abaixo sugerem que você esteja usando uma rede com os ips 192.168.0... e a interface de rede interna é eth1.

```
# iptables -t nat -A PREROUTING -i
eth1 -s 192.168.0.0/24 -p tcp --dport
80 -j REDIRECT --to-port 3180
# iptables -t nat -A PREROUTING -i eth1
-s 192.168.0.0/24 -p tcp --dport 443 -j
REDIRECT --to-port 31443
```

Continuando...

Passo 15) Parar o squid. Em sistema Debian e derivados,

pode ser usado também o comando /etc/init.d/squid3 stop ou service squid3 stop.

```
# systemctl stop squid3
```

Passo 16) Entrar no diretório de configurações do Squid.

```
# cd /etc/squid3
```

Passo 17) Criar os diretórios "cert" e "list". Por favor, não coloque aspas quando for digitar os nomes!

```
# mkdir cert list
```

Passo 18) Definir as permissões dos diretórios criados no passo anterior. Se você tiver ilusões de perseguição ou cultivar teorias de conspiração, mude o proprietário e permissões dos diretórios agora. No geral, não há muito com que se preocupar fora dos filmes de ficção científica...

```
# chmod -R 777 cert/ list/
```

Passo 19) Entrar no diretório "cert".

```
# cd cert
```

Passo 20) Criar a chave privada do certificado digital. No lugar de "servidor.com" você pode colocar o seu domínio,

se tiver. Preste atenção, se quiser colocar o seu domínio real no certificado, deixe a extensão .key no final. Por exemplo, seudominio.com, o arquivo ficaria seudominio.com.key. Para escrever o artigo, retiramos nosso domínio real e usamos "servidor.com". Para informações detalhadas, no final do artigo você encontra as fontes de pesquisa.

```
# openssl genrsa -out servidor.com.key 2048
```

Passo 21) Criar o CSR (Certificate Signing Request). Observe que logo depois do comando o openssl vai pedir as informações, que você deve preencher corretamente, pois o certificado pode não funcionar sem elas. Na ordem, cada linha: País, Estado, Cidade, Empresa ou Seu Nome, Setor da Empresa (pode ser só Ti), Common Name é onde você vai colocar seu domínio ou pode deixar servidor.com; por último seu e-mail. Parece muito óbvio mas vamos lembrar: depois de cada linha que você preencher tecle Enter! No exemplo abaixo, alteramos as informações originais, então preste atenção no que cada linha pede, depois de : (dois pontos). Se preferir, na linha Common Name, ao invés de servidor.com, coloque o ip do seu servidor, por exemplo, 192.168.0.1. É isso que vai identificar para os clientes da rede quem está gerando o certificado que vai assegurar a comunicação.

```
# openssl req -new -key
servidor.com.key -out servidor.com.csr

Country Name (2 letter code) [AU]: BR
State or Province Name (full name) [Some-
State]: Seu Estado
Locality Name (eg, city) [ ]: Sua Cidade
Organization Name (eg, company) [Internet
Widgits Pty Ltd]: Sua Empresa ou Seu
Nome
Organizational Unit Name (eg, section) [
]: Ti
Common Name (e.g. server FQDN or YOUR
name) [ ]: servidor.com
Email Address [ ]: seuemail
A challenge Password [ ]: <<<<<< Deixe em
branco, só aperte Enter
An optional company name [ ]: <<<<<< Deixe
em branco, só aperte Enter
```

Passo 22) Assinar o certificado e gerar um arquivo CRT, que é a chave pública. No final indicamos um artigo que explica melhor como tudo funciona. Aconselhamos a leitura.

```
# openssl x509 -req -days 3650 -in
servidor.com.csr -signkey
```

```
servidor.com.key -out servidor.com.crt
```

Passo 23) Juntar a chave privada e a chave pública em um arquivo de extensão PEM. O Squid precisa desse arquivo para funcionar corretamente.

```
# cat servidor.com.key
servidor.com.crt > servidor.com.pem
```

Passo 24) Copiar o arquivo CRT para um local onde possa ser disponibilizado para os clientes. Alguns sites não abrem pois dá erro de certificado, então você importa esse CRT para o navegador e pronto! Como instalamos o apache (servidor web), podemos facilitar a vida:

```
# mkdir /var/www/html/cert
# cp servidor.com.crt /var/www
/html/cert
```

Importar o Certificado para o computador cliente no Chrome:

1. No navegador, digite o endereço do certificado (192.168.0.1/cert) e, quando abrir, clique sobre o arquivo (servidor.com.crt). Isso vai salvar o certificado no computador cliente;

2. Abra as configurações do Chrome. Em cima, a direita,

tem 3 traços, clique sobre eles;

3. Vá para o final das configurações, em baixo, e clique na frase azul "Mostrar configurações avançadas...";

4. Clique no botão "Gerenciar Certificados";

5. Vá para a aba "Autoridades de Certificação Raiz Confiáveis"; <<<<< Se não for nessa aba, o certificado não será usado

6. Clique em "Importar", "Avançar", "Procurar", localize o certificado que foi baixado, clique sobre ele e clique em Abrir, depois Avançar, Avançar, Concluir, Sim, Ok e fechar.

Importar o Certificado para o computador cliente no Firefox:

1. Digite o endereço do certificado (192.168.0.1/cert) e clique sobre o arquivo;

2. Na janela que abrir, marque as 3 caixas, depois clique em OK.

O Internet Explorer usa o certificado importado pelo Chrome. Outros navegadores também podem usar o certificado importado pelo Firefox. De qualquer forma é fácil, se tiver dificuldades, comente que ajudamos.

CONFIGURAÇÃO DO SQUID - PARTE 2

Passo 25) Entrar no diretório list, onde vamos criar arquivos simples. Nesses arquivos, em cada linha, colocamos um site que será liberado ou bloqueado. Também é possível colocar ips e outras informações que serão usadas pelo Squid, de acordo com a "regra" configurada. Veremos isso mais a frente. Observe que estamos no diretório cert (caminho completo: /etc/squid3/cert), então o comando vai lhe trazer para o diretório acima (squid3), depois entrar no diretório list.

```
# cd ../list
```

Passo 26) Criar os arquivos, chamados sitessl, sitehttp, sitebloq, palavras e pcfree.

```
# touch sitessl sitehttp sitebloq
palavras
```

Passo 27) Inserir endereços (um em cada linha) de sites com SSL (HTTPS) que serão sempre liberados no arquivo sitessl, endereços de sites que NÃO usam SSL e serão sempre liberados no arquivo sitehttp, endereços de sites que serão sempre bloqueados independente dos esforços

do usuário em burlar seu sistema no arquivo sitebloq, coloque palavras que podem ser encontradas em sites que precisam ser bloqueadas no arquivo palavras, por fim coloque um ou mais ips de equipamentos na sua rede que pode(m) ter acesso irrestrito.

ATENÇÃO! No arquivo "palavras", você pode colocar apenas palavras que podem estar na URL (endereço do site), não é necessário colocar a URL completa, mas pode colocá-las também. O Squid usará estes arquivos para saber se o site/equipamento está liberado para acesso ou não. Observe que alguns sites são colocados em mais de uma lista, pois eles podem usar tanto a porta 80 (HTTP) quanto a 443 (HTTPS) e também prevemos a alteração da url, então colocamos o domínio na lista "palavras". O Squid trata de forma diferente cada protocolo, no caso do HTTPS é feito um processo chamado SSL Bump, enquanto para HTTP o controle é mais fácil.

LEIA os artigos indicados ao final para entender o Bump, assim poderá gerenciar melhor seu servidor. Parece redundante criar uma lista com sites liberados e depois outra lista com sites bloqueados, mas a ideia é liberar apenas alguns sites específicos, deixar bem definido quais serão bloqueados sempre e bloquear todo o resto que for solicitado pela rede. Assim você ganha flexibilidade para liberar só o que precisa e bloquear todo o resto.

Abaixo, apenas exemplos; só para liberar o GMail sem nenhum erro, foi necessário liberar 6 (seis) domínios; mude os sites para a sua necessidade:

```
# vim sitessl
```

```
accounts.google.com
accounts.google.com.br
gmail.com
mail.google.com
googleusercontent.com
gstatic.com
hotmail.com
imp.live.com
login.live.com
mail.live.com
```

```
# vim sitehttp
```

```
empresa.com
outrosite.com
gmail.com
hotmail.com
```

```
# vim sitebloq
```

```
facebook.com
youtube.com
```

```
# vim palavras
```

```
porno
video
musica
jogos
facebook.com
youtube.com
```

```
# vim pcfree
```

```
192.168.0.101
192.168.0.102
```

Passo 28) Criar um diretório que armazenará os certificados temporários, usados a cada conexão que o Squid gerenciar. Seria um banco de dados de certificados. Para gerenciar esse banco de dados, o Squid usa a biblioteca ssl_crtd.

```
# /usr/lib/squid3/ssl_crtd -c -s
```

```
/var/lib/ssl_db/
```

Passo 29) Mudar a permissão do diretório criado no passo anterior, para que o Squid possa usá-lo a vontade.

```
# chown -R proxy /var/lib/ssl_db/
```

Passo 30) Voltando para o diretório /etc/squid3. Lembre que foi onde criamos os diretórios cert e list. Como entramos nos diretórios criados, agora voltamos para um nível acima.

```
# cd ..
```

Passo 31) Criar uma cópia do arquivo de configuração original do Squid e limpá-lo para inserir nossa configuração personalizada.

```
# cp squid.conf squid.conf.bkp
# echo "" > squid.conf
```

Passo 32) Abrir o arquivo de configuração e colar as linhas abaixo. No exemplo, estamos gerenciando uma rede com ip de classe C (192.168.0.0). Por padrão, o Squid usa porta 3128, bloqueamos esta porta com firewall, questão de segurança, isto seria assunto para outro Artigo. Mesmo assim o Squid precisa manter a porta 3128 para fazer um "forward interno"... Direcionamos pelo Firewall as conexões da porta 80 para a porta 3180 e da porta 443 para a 31443.

Aconselhamos a leitura dos artigos sobre "man-in-the-middle" e "ssl_bump".

 # vim squid.conf

```
##> Enderecos e Portas
http_port 3128
http_port 3180 intercept
https_port 31443 intercept ssl-bump generate-host-certificates=on
dynamic_cert_mem_cache_size=4MB cert=/etc/squid3/cert/servidor.com.pem

##> Adm
shutdown_lifetime 3 seconds
cache_mgr Administrador
cache_effective_user proxy
httpd_suppress_version_string on

##> Cache e Memoria
cache_mem 1024 MB
maximum_object_size_in_memory 10240 KB
memory_replacement_policy lru
cache_replacement_policy lru
maximum_object_size 100 MB
cache_dir ufs /var/spool/squid3 102400 16 256
coredump_dir /var/spool/squid3
```

```
store_dir_select_algorithm least-load
max_stale 1 week

##> Logs
access_log daemon:/var/log/squid3/access.log proxy
logfile_daemon /usr/lib/squid3/log_file_daemon
cache_log /var/log/squid3/cache.log

##> Gerenciamento de pacotes
quick_abort_min 16 KB
quick_abort_max 16 KB
quick_abort_pct 95
negative_ttl 0 seconds
positive_dns_ttl 6 hours
negative_dns_ttl 1 minutes
client_request_buffer_max_size 512 KB

##> Performance
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i \.(gif|png|jpg|jpeg|ico)$ 10080 90%
43200 override-expire ignore-no-store ignore-private
refresh_pattern -i
\.(iso|avi|wav|mp3|mp4|mpeg|swf|flv|x-flv)$ 43200 90%
432000 override-expire ignore-no-store ignore-private
refresh_pattern -i
\.(deb|rpm|exe|zip|tar|tgz|ram|rar|bin|ppt|doc|tiff)$
10080 90% 43200 override-expire ignore-no-store
ignore-private
```

```
refresh_pattern -i \.index.(html|htm)$ 0 40% 10080
refresh_pattern -i \.(html|htm|css|js)$ 1440 40% 40320
refresh_pattern . 0 20% 4320
#Videos
refresh_pattern -i youtube.com/.* 10080 90% 43200
refresh_pattern (/cgi-bin/|\?) 0 0% 0

##> Diretivas para SSL
sslcrtd_program /usr/lib/squid3/ssl_crtd -s /var/lib
/ssl_db -M 4MB
sslcrtd_children 32 startup=5 idle=1
sslcrtvalidator_children 32 startup=5 idle=1
concurrency=1
dead_peer_timeout 10 seconds
forward_max_tries 10

##> ACLs Iniciais
acl SSL_ports port 443
acl Safe_ports port 443 80
acl CONNECT method CONNECT

##> ACLs Personalizadas
acl localnet src 192.168.0.0/24
acl PCLiberado src "/etc/squid3/list/pcfree"
acl SiteSeguroLiberado dstdomain "/etc/squid3
/list/sitessl"
acl SiteNaoSeguroLiberado dstdomain "/etc/squid3
/list/sitehttp"
```

```
acl SiteBloqueado dstdomain "/etc/squid3/list/sitebloq"
acl PalavraBloqueada url_regex -i "/etc/squid3
/list/palavras"
acl Musica url_regex -i .mp3$ .wma$ .wav$ .ogg$
.mp4$
acl Virus url_regex -i .inf$ .vbs$ .irc$ .src$ .bat$ .cmd$
.reg$ .exe$
acl Video url_regex -i .avi$ .rmvb$ .flv$ .mpeg$ .3gp$
.4gp$ .3g2$ .mkv$

##> Acessos Iniciais
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager

##> Acessos Personalizados
http_access allow localhost
http_access allow SiteSeguroLiberado
http_access allow SiteNaoSeguroLiberado
http_access allow PCLiberado
http_access deny SiteBloqueado
http_access deny PalavraBloqueada
http_access deny Musica Virus Video
http_access deny localnet
http_access deny all

##> Acessos SSL
```

```
sslproxy_flags DONT_VERIFY_PEER
sslproxy_cert_error deny !SiteSeguroLiberado
always_direct deny !SiteSeguroLiberado
ssl_bump none localhost
ssl_bump none SiteSeguroLiberado PCLiberado
ssl_bump server-first all
```

Passo 33) O Squid pode criar os diretórios de Cache, onde serão armazenados os arquivos de sites que passam por ele. Existe uma organização específica para esses diretórios, que conseguimos com o comando:

```
# squid3 -z
```

Passo 34) Enfim, iniciar nosso Squid.

```
# systemctl start squid3
```

Passo 35) Verificar se o Squid foi iniciado corretamente. Depois do comando abaixo, se aparecer as linhas loaded, active, started... está tudo certo. Veja que colocamos vários X para substituir números que vão mudar de acordo com seu sistema.

```
# systemctl status squid3
squid3.service - LSB: Squid HTTP Proxy
version 3.x
```

```
   Loaded: loaded (/etc/init.d/squid3)
   Active: active (running) since Seg
2016-02-XX XX:XX:XX UTC; XXs ago
  Process: XXXX ExecStop=/etc/init.d
/squid3 stop (code=exited,
status=0/SUCCESS)
  Process: XXXX ExecStart=/etc/init.d
/squid3 start (code=exited,
status=0/SUCCESS)
   CGroup: /system.slice/squid3.service
           XXXX /usr/sbin/squid3 -YC -f
/etc/squid3/squid.conf
           XXXX (squid-1) -YC -f
/etc/squid3/squid.conf
           XXXX (ssl_crtd) -s /var/lib
/ssl_db -M 4MB -b 4096
           XXXX (ssl_crtd) -s /var/lib
/ssl_db -M 4MB -b 4096
           XXXX (ssl_crtd) -s /var/lib
/ssl_db -M 4MB -b 4096
           XXXX (ssl_crtd) -s /var/lib
/ssl_db -M 4MB -b 4096
           XXXX (ssl_crtd) -s /var/lib
/ssl_db -M 4MB -b 4096
           XXXX (logfile-daemon) /var/log
```

```
/squid3/access.log
           XXXX (unlinkd)
           XXXX (pinger)

Fev XX XX:XX:XX servidor squid3[XXXX]:
Squid Parent: will start 1 kids
Fev XX XX:XX:XX servidor squid3[XXXX]:
Starting Squid HTTP Proxy 3.x: squid3.
Fev XX XX:XX:XX servidor squid3[XXXX]:
Squid Parent: (squid-1) process XXXX
started
```

REDE E FIREWALL

Passo 36) Abaixo, configurações ideais para o arquivo hosts. É ele que identifica seu servidor na rede e é usado também por alguns processos internos quando precisam identificar o servidor. Lembre-se que estamos usando como exemplo uma rede com ips 192.168.0....

```
# vim /etc/hosts
```

```
127.0.0.1   localhost.servidor.com       l

192.168.0.1   servidor.servidor.com      s
192.168.0.1   ns1.servidor.com           r

127.0.1.1   servidor.servidor.com        s
127.0.1.1   ns1.servidor.com             r

# The following lines are desirable for IP

::1     localhost ip6-localhost ip6-loopba
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Passo 37) Configurar as interfaces de rede. Para quem veio do "mundo das janelas" seria o equivalente a clicar naquele computadorzinho ao lado do relógio e mudar o ip... No nosso exemplo, recebemos ip da internet por DHCP na interface eth0 e definimos para a eth1 o ip 192.168.0.1. O servidor tem duas placas de rede, uma onboard (eth0) e outra offboard (eth1). Identificamos isso com o comando ifconfig -a.

 # vim /etc/network/interfaces

```
iface lo inet loopback
```

```
#Interface Internet
iface eth0 inet dhcp

#Interface Rede Local
iface eth1 inet static
address 192.168.0.1
netmask 255.255.0.0
broadcast 192.168.255.255
network 192.168.0.0

allow-hotplug eth0

allow-hotplug eth1

auto lo

auto eth0

auto eth1
```

Passo 38) Configurar os servidores DNS que atenderão nosso servidor. Mais a frente indicamos uma configuração para ele mesmo ser um servidor DNS, o que deve melhorar bastante o desempenho da sua rede.

```
# vim /etc/resolv.conf
```

```
search servidor.com
nameserver 192.168.0.1
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Passo 39) Reinicia a rede. Em alguns casos a máquina não reconhece as configurações de rede só reiniciando o serviço, então faça um reboot.

```
# systemctl restart networking
```

Passo 40) Criar um script de firewall. Copie todo o texto abaixo, desde a linha #!/bin/bash até a linha exit0 e cole no seu arquivo. Lembre-se que para começar a inserir texto no VIM, digite a letra i. Pra colar vai depender de como estiver acessando. Geralmente por SSH, clicar com o botão auxiliar (direito) do mouse vai colar.

```
# vim /etc/init.d/firewall
```

```
#!/bin/bash

PATH=/sbin:/bin:/usr/sbin:/usr/bin

### BEGIN INIT INFO
# Provides:          firewall
# Required-Start:    $local_fs $remote_fs $netw
# Required-Stop:     $local_fs $remote_fs $netw
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start Firewall at boot tir
# Description:       Enable service provided by
### END INIT INFO

export TX="echo -e"

default(){
$TX ""
$TX ">>>>>>>>>>>>>>>>>> Iniciando Script de Fi

$TX "Exportar Variaveis....................
export IFNET=eth0
export IFLOC=eth1
export MP=/sbin/modprobe
export RMM="$(which rmmod)"
export IPT=/sbin/iptables
export SVC=/etc/init.d
export IP4=/proc/sys/net/ipv4
export IP4C=/proc/sys/net/ipv4/conf
export NETW=192.168.0.0/24
$TX "......................................
}

start() {
```

```
echo "Definir politicas padrao..............
$IPT -P INPUT DROP
$IPT -P FORWARD DROP
$IPT -P OUTPUT DROP
$TX "....................................

$TX "Desbloquear RP_Filter..................
if [ -e $IP4C/all/rp_filter ]; then
 for f in $IP4C/*/rp_filter; do
$TX "0" > $f
done
fi
$TX "....................................

$TX "Bloquear tudo que for desnecessário.....
$IPT -p tcp --dport 3128 -j DROP
$IPT -A INPUT -m iprange --src-range 192.168.1
$TX "....................................

$TX "Direcionar para Proxy-Cache.............
$IPT -t nat -A PREROUTING -i $IFLOC -s $NETW -p
$IPT -t nat -A PREROUTING -i $IFLOC -s $NETW -p
$TX "....................................

$TX "Compartilhar conexao...................
$IPT -t nat -A POSTROUTING -o $IFNET -j MASQUER
if [ -e $IP4/ip_forward ]; then
$TX "1" > $IP4/ip_forward
else
$TX "Erro ao habilitar Forward!! Falha Grave no
fi
$IPT -A FORWARD -s $NETW -j ACCEPT
$IPT -A FORWARD -d $NETW -j ACCEPT
$TX "....................................
```

```
$TX "Permitir Trafegos de entrada e Saida.....
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A INPUT -s $NETW -j ACCEPT
$IPT -P OUTPUT -j ACCEPT
$TX ".....................................

$TX " "
$TX ">>>>>>>>>>>>>>>>>>>>> Firewall Iniciado!
}

stop(){
$TX "Limpar todas as  regras de firewall......
export IPT=/sbin/iptables
export SVC=/etc/init.d
$IPT -X
$IPT -F
# Zerar contadores
$IPT -Z
# Definir politicas padrao
$IPT -P INPUT DROP
$IPT -P FORWARD DROP
$IPT -P OUTPUT DROP
$TX ".....................................

$TX "Desfazer Compartilhamento.............
# Desativar trafego entre as placas de rede
$TX 0 > $IP4/ip_forward
# Ativar rp_filter
if [ -e $IP4C/all/rp_filter ]; then
 for f in $IP4C/*/rp_filter; do
$TX "1" > $f
done
# Reiniciar servico de rede
```

```
$SVC/networking restart
fi
$TX ".....................................

$TX ">>>>>>>>>>>>>>>>>>>>>> Firewall Parado!!!
}

case "$1" in

        inicio) while true; do
                default
                break
                done
;;

        start) while true; do
                default
                start
                break
                done
;;

        stop) while true; do
                stop
                break
                done
;;

        restart) while true; do
                stop
                default
                start
                break
                done
```

```
;;

esac

exit 0
```

Passo 41) Fazer com que o script de firewall seja executável e inicie junto com o sistema.

```
# chmod +x /etc/init.d/firewall
# update-rc.d firewall defaults
```

Passo 42) Iniciar o firewall.

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

Passo 43) Verificar se as regras foram aplicadas. Observe que no segundo comando está aparecendo os redirecionamentos de portas. Apagamos as nossas regras, mas você vai perceber em cada sessão, abaixo das palavras target, prot, opt, source e destination, informação do que o firewall está fazendo. É fácil entender, no exemplo abaixo, o alvo (target) é um redirecionamento (REDIRECT), usando protocolo (prot) TCP, nenhuma opção adicional (opt: --), onde a origem (source) é todos (anywhere), então o destino (destination) será todos (anywhere), protocolo TCP, porta HTTP (80) e HTTPS (443), redirecionando para as portas 3180 e 31443, respectivamente.

```
# iptables -L

Chain PREROUTING (policy ACCEPT)
target     prot opt source               desti

Chain INPUT (policy ACCEPT)
target     prot opt source               desti

Chain OUTPUT (policy ACCEPT)
target     prot opt source               desti

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               desti


# iptables -t nat -L

Chain PREROUTING (policy ACCEPT)
target     prot opt source               desti
REDIRECT   tcp  --  anywhere             anywh
REDIRECT   tcp  --  anywhere             anywh

Chain INPUT (policy ACCEPT)
target     prot opt source               desti

Chain OUTPUT (policy ACCEPT)
target     prot opt source               desti

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               desti
```

DNS E DHCP

Passo 44) Parar o servidor DNS.

```
# systemctl stop bind9
```

Passo 45) Abrir um arquivo para fazer configuração personalizada do seu servidor. Observe que abaixo do comando colocamos o conteúdo que você vai copiar daqui e colar no servidor. Mude os nomes e ips, caso seja necessário. Abaixo, nosso exemplo usa o domínio "servidor.com" (Sem aspas, por favor!) e 192.168 ao contrário (168.192). Se o seu for diferente, mude.

```
# vim /etc/bind/named.conf.local
```

```
zone "servidor.com" {
type master;
file "/etc/bind/db.servidor.com";
};

zone "168.192.in-addr.arpa"{
type master;
```

```
file "/etc/bind/db.168.192";
};
```

Passo 46) Criar e já abrir o primeiro arquivo. Abaixo o conteúdo dele. Basta copiar e colar, verificando se tem alguma diferença para o seu caso. Preste muita atenção pois qualquer coisa errada fará seu servidor não funcionar. De preferência, copie o texto abaixo e mude apenas o que for necessário.

 # vim /etc/bind/db.servidor.com

```
$TTL 604800
@       IN      SOA
        2013120400                  ; Seri
        604800                      ; Refr
        86400                       ; Retr
        2419200                     ; Expi
        604800  )                   ; Nega
;

servidor.com.               A
servidor.servidor.com.    A
ns1.servidor.com.                   A

www                                 CNAME
```

```
# vim /etc/bind/db.168.192
```

```
$TTL 604800
@       IN      SOA                     ns
        2013120400              ; Serial
        604800                  ; Refresh
        86400                   ; Retry
        2419200                 ; Expire
        604800  )               ; Negative
;
                NS
1.0             PTR
1.0             PTR
1.0             PTR
```

Passo 47) Iniciar o servidor DNS.

```
# systemctl start bind9
```

Passo 48) Testar o servidor DNS. Se os resultados dos comandos ficarem parecidos com as linhas abaixo, deu certo. Observe que o exemplo pode ser diferente do que aparece no seu servidor se você usou informações diferentes, como nome de host, ip, domínio, etc. Colocamos aqui vários comandos para testes, aconselhamos que você use todos.

```
# host servidor.com
servidor.com has address 192.168.0.1

# host 192.168.0.1
1.0.168.192.in-addr.arpa domain name
pointer ns1.servidor.com.
1.0.168.192.in-addr.arpa domain name
pointer servidor.com.
1.0.168.192.in-addr.arpa domain name
pointer servidor.servidor.com.

# nslookup 192.168.0.1
 Server:         192.168.0.1
 Address:        192.168.0.1#53

 1.0.168.192.in-addr.arpa name = servidor.com.
 1.0.168.192.in-addr.arpa name = ns1.servidor.c
 1.0.168.192.in-addr.arpa name = servidor.servi


# nslookup servidor.com
 Server:         192.168.0.1
 Address:        192.168.0.1#53

 Name:   servidor.com
 Address: 192.168.0.1
```

```
# dig 192.168.0.1
; <<>> DiG 9.9.5-9+deb8u5-Debian <<>>
192.168.0.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status:
NXDOMAIN, id: XXXXX
;; flags: qr rd ra ad; QUERY: 1, ANSWER:
0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;192.168.0.1.                    IN      A

;; AUTHORITY SECTION:
.                       10800
IN      SOA     a.root-servers.net.
nstld.verisign-grs.com. 2016012801 1800
900 604800 86400

;; Query time: XXX msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: XXX Fev XX XX:XX:XX BRT 2016
;; MSG SIZE  rcvd: XXX
```

```
# dig servidor.com
; <<>> DiG 9.9.5-9+deb8u5-Debian <<>>
servidor.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status:
NOERROR, id: XXXX
;; flags: qr aa rd ra; QUERY: 1, ANSWER:
1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;servidor.com.                  IN      A

;; ANSWER SECTION:
servidor.com.           604800  IN      A
192.168.0.1

;; AUTHORITY SECTION:
servidor.com.           604800  IN      NS

;; Query time: 0 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
```

```
;; WHEN: XXX Fev XX XX:XX:XX BRT 2016
;; MSG SIZE  rcvd: XX
```

Passo 49) Parar o servidor DHCP.

```
# systemctl stop isc-dhcp-server
```

Passo 50) Entrar no diretório de configurações do DHCP, fazer uma cópia do arquivo de configuração e limpar o original para inserir o que precisamos.

```
# cd /etc/dhcp
# cp dhcpd.conf dhcpd.conf.bkp
# echo "" >dhcpd.conf
```

Passo 51) Abrir o arquivo conf e colar as linhas abaixo. Veja que é uma configuração básica, para qualquer equipamento que se conecte na rede receber ip automaticamente. Se preferir um pouco mais de segurança, tire a # (cerquilha, jogo da velha, grade, quadrado, como você quiser chamar...) no começo de todas as linhas que tiver, e adicione cada máquina da sua rede no grupo (usamos o nome do grupo "exemplo", então você pode mudar como quiser, apenas essa palavra). Se entender bem, pode até criar outros grupos, mas observe que uma linha errada fará com que o DHCP não funcione!

Veja que usamos o nome do host "teste", isso você precisa

mudar, na linha hardware ethernet você coloca o MAC/Endereço Físico de cada máquina e na linha fixed-address vai o ip que a máquina sempre vai receber. Lembre-se de tirar a # (comentário) da primeira linha! Esse trabalho pode ser cansativo se a rede for grande, mas existem programas que fazem isso mais facilmente.

Outra possibilidade é você verificar no próprio DHCP server qual ip foi cedido para cada máquina e usar isso para fixar a configuração, sem precisar ir de máquina em máquina anotando o MAC. Não tem jeito! Se quiser segurança precisa ter trabalho! Vai precisar digitar cada nome de host, cada endereço físico e cada ip, individualmente! Dica: Estagiários adoram esse tipo de serviço... "No Pain, No Gain...".

```
# vim /etc/dhcp/dhcpd.conf
```

```
#deny client-updates;
deny unknown-clients;
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;
log-facility local7;
ddns-updates off;
one-lease-per-client false;
```

```
deny bootp;
option dhcp-server-identifier servidor.com;
option domain-name "servidor.com";
option routers 192.168.0.1;
option broadcast-address 192.168.0.255;
option domain-name-servers 192.168.0.1, 8.8.8.8,
8.8.4.4;

subnet 192.168.0.0 netmask 255.255.255.0 {

range 192.168.0.2 192.168.0.254;

#group exemplo {
#host teste {
#hardware ethernet XX:XX:XX:XX:XX:XX;
#fixed-address 192.168.X.X;
#}

}
```

Passo 52) Iniciar o servidor DHCP.

```
# systemctl start isc-dhcp-server
```

Passo 53) Verificar se está funcionando.

```
# systemctl status isc-dhcp-server
```

```
dhcps.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-
server)
   Active: active (running) since XXX
2016-02-XX XX:XX:XX BRT; Xh XXmin ago
  Process: XXXXX ExecStart=/etc/init.d
/dhcps start (code=exited,
status=0/SUCCESS)
   CGroup: /system.slice/dhcps.service
           XXXXX /usr/sbin/dhcpd -q -cf
/etc/dhcp/dhcpd.conf -pf /var/run
/dhcpd.pid eth1
```

Passo 54) Verificar quais ips já foram servidos. Pode demorar um pouco até mostrar resultado, é o tempo em que as máquinas da sua rede forem recebendo os ips. Se você deixar todos "amarrados" no DHCP, com MAC e IP registrados, esse comando não mostrará resultado.

```
# cat /var/lib/dhcp/dhcpd.leases
```

Passo 55) É sempre melhor você reiniciar o servidor para ter certeza de que tudo vai iniciar corretamente. O ideal é que o servidor nunca seja desligado, mas podem haver falhas de energia, fontes queimadas... muitos motivos para ser necessário reiniciar o sistema.

```
# reboot
```

CONCLUSÃO E LINKS

Como dissemos, ficou bem longo! O importante é que, se você seguir todos os passos corretamente, não terá a mesma frustração que tivemos por meses, precisando de um negócio que é a melhor solução mas parecia nunca funcionar!

Nosso servidor funciona perfeitamente e agora temos tranquilidade, pois alguns usuários vivem procurando formas de acessar conteúdos indevidos.

Com esse Squid, basta liberar só o que você deixar, o resto será bloqueado, sem choro...

Seguem as fontes de pesquisa:

- http://debianbase.blogspot.com.br/2015/05/installsquid34withsslbumpon.html (http://debianbase.blogspot.com.br/2015/05/installsquid34withsslbumpon.html)
- http://codepoets.co.uk/2014/squid34xwithsslfordebianwheezy/

(http://codepoets.co.uk
/2014/squid34xwithsslfordebianwheezy/)
- [SOLVED] Problems rebuilding squid3 with ssl (http://ubuntuforums.org /showthread.php?t=2171061)
- Filtro de Conteúdo Web (Proxy) - Marco Túlio Oliveira de Moraes (https://sites.google.com /site/matulio/artigos/proxy)
- Administrator's Guide 4.4 (Alpha 2) — Diladele Web Safety 4.3.0 documentation (http://docs.diladele.com/administrator_guide_4_4 /index.html)
- http://www.panticz.de /SquidCompilewithSSLsupportunderDebianJessie (http://www.panticz.de /SquidCompilewithSSLsupportunderDebianJessie)
- http://www.squidcache.org/Versions/v3/3.4/cfgman /cache_peer.html (http://www.squidcache.org /Versions/v3/3.4/cfgman/cache_peer.html)
- http://www.squidcache.org/Versions/v3/3.4/cfgman /https_port.html (http://www.squidcache.org /Versions/v3/3.4/cfgman/https_port.html)
- http://www.squidcache.org/Versions/v3/3.4/cfgman /ssl_bump.html (http://www.squidcache.org/Versions /v3/3.4/cfgman/ssl_bump.html)

- http://www.squidcache.org/mailarchive/squidusers/201303/0046.html (http://www.squidcache.org/mailarchive/squidusers/201303/0046.html)
- http://www.squidcache.org/Doc/config/ssl_bump/ (http://www.squidcache.org/Doc/config/ssl_bump/)
- http://wiki.squidcache.org/Features/SslBump (http://wiki.squidcache.org/Features/SslBump)
- http://wiki.squidcache.org/ConfigExamples/Intercept/DebianWithRedirectorAndReporting (http://wiki.squidcache.org/ConfigExamples/Intercept/DebianWithRedirectorAndReporting)
- http://wiki.squidcache.org/ConfigExamples/Intercept/SslBumpExplicit (http://wiki.squidcache.org/ConfigExamples/Intercept/SslBumpExplicit)
- http://www.squidcache.org/Doc/config/refresh_pattern/ (http://www.squidcache.org/Doc/config/refresh_pattern/)
- squid 3.5 peek and splice how to (http://marek.helion.pl/install/squid.html)
- http://www.linuxers.com.br/bloquearfacebooksquidiptableserotas/ (http://www.linuxers.com.br/bloquearfacebooksquidiptableserotas/)
- Códigos de status do squid (http://www.savant.com.br/index.php/artigos

/tutoriais/34)

- http://blog.manty.net/2014/12/squidproxybeingtransparentalsofor.html (http://blog.manty.net/2014/12/squidproxybeingtransparentalsofor.html)
- Linux.com :: Speed up your Internet access using Squid's refresh patterns (http://archive09.linux.com/feature/153221)
- http://squidwebproxycache.1019090.n4.nabble.com/BypassingSSLBumpfordstdomaintd4658835.html (http://squidwebproxycache.1019090.n4.nabble.com/BypassingSSLBumpfordstdomaintd4658835.html)
- Squid 3.1 Caching Proxy with SSL | thejimmahknows (http://thejimmahknows.com/squid-3-1-caching-proxy-with-ssl/)
- https://aacable.wordpress.com/2013/10/22/squid3andssl/ (https://aacable.wordpress.com/2013/10/22/squid3andssl/)

Voltar (verArtigo.php?codigo=15810)