Manual Servidor Debian

Divulgar Linux
é
Divulgar
Conhecimento

Alexandre Stürmer Wolf

Sumário

Introdução

Esse tutorial é uma compilação revisada de materiais, juntamente com experiências e experimentos do autor. A compilação de materiais é baseada nos sites da internet que de alguma forma refere-se a comunidade Debian Linux. A maior fonte de informação é obtida em www.debian.org.

O tutorial não tenta de forma alguma abranger outras distribuições, no entanto, Linux é Linux, assim muitas coisas são iguais ou equivalentes entre as distribuições.

Muitos conteúdos desse tutorial foram *copiados* e *adaptados* de sites "bem conhecidos" da internet. Atualmente existe muito material sobre os assuntos abordados, mas poucos validados e realmente experimentados. Esse material foi e é escrito observando as dúvidas e dificuldades de usuários Debian em um laboratório específico para essa finalidade, onde os usuários possuem os mais variados níveis de conhecimento.

Todo e qualquer conteúdo ou correção são aceitas, pois esse é um tutorial livre desenvolvido pela comunidade Debian e compilado por Alexandre Stürmer Wolf (as_wolf@terra.com.br). O material foi escrito utilizando-se de ferramentas livres, mais precisamente o editor do OpenOffice (www.openoffice.org) e a distribuição Debian Linux (www.debian.org). Todas as distribuições são excelentes, depende do conhecimento de cada um para torná-la mais adequada às suas necessidades.

Sobre o autor

Alexandre Stürmer Wolf é desenvolvedor de sistemas comerciais e livres, possui repugnância ao uso de softwares piratas pelas *empresas*, acredita em software livre nas Universidades e alguns órgãos públicos, desde que não entrem em conflitos com o desenvolvimento mercadológico (irônica e contraversiva nessa frase). Não possui nada contra os produtos Windows, pior de tudo é que possui inveja do império Microsoft (adoraria possuir milhões). Atualmente utiliza Debian (para tudo) e o Windows 98 (desenvolvimento comercial em Delphi IV (possui o registro, gosto do S.O., desde que não precise alterar configurações de rede)), possui ainda uma instalação do Windows XP em sua máquina para tirar as dúvidas de clientes.

Trabalha também como professor Universitário nas áreas de engenharias e informática, é técnico em análises químicas, gemologia, eletrônica, Bacharel em Análises de Sistemas, Mestre em Engenharia Elétrica, e continua estudando para as próximas titulações. Desenvolvedor atuante em Delphi, Java, Lazarus, C's, Clipper, Free Pascal, Assembly, entre outras várias linguagens menos comuns.

Agradecimentos

Primeiramente à minha esposa Merlin, por aturar a mim e meus aeromodelos, fora a bagunça de componentes eletrônicos espalhados pela casa. A paciência das turmas de laboratório de experimentação pela demora na entrega desse material. A comunidade Debian principalmente, e todas as outras comunidades as quais pertenci e tenho grande estima (Slackware e Red Hat).

Antes de começar

A pergunta fatal, "qual a melhor distribuição", hoje digo Debian, mas a resposta correta é aquela que mais lhe agradar, pois Linux é Linux. A diferença entre distribuições deixou de ser apenas "quais pacotes" fazem parte ou não, atualmente cada distribuição *Raiz*, possui mecanismos que facilitam a vida do usuário final de alguma forma, seja instalação de aplicativos, esquema de diretórios, "frescuras", e tudo mais. Hoje, indiferente da distribuição os pacotes já se encontram traduzidos para muitas línguas, caso ainda não esteja, você mesmo pode fazer isso, ou tenha um pouco de paciência.

Algumas pessoas afirmam que a Debian é uma distribuição estável mas *atrasada*, discordo plenamente dessa afirmação, pois a mesma oferece mecanismos de forma a utilizar programas estáveis, ou se desejar, é possível utilizar o "último grito" em programas. Por exemplo, nesse momento estou usando o último Kernel, a última versão do OpenOffice, e o mesmo para outros programas. Nas empresas quando instalo o Linux, principalmente servidores, somente utilizo a versão *stable*, que pode ser um pouco desatualizada em relação aos últimos lançamentos, mas é garantida. Já em casa, uso os últimos lançamentos, para testá-los e estar a par das novas características (coisa de "micreiro").

Espero que esse tutorial possa ajudar, favor distribuir para o maior número de pessoas. Divulgar Linux é divulgar conhecimento, quanto mais pessoas o utilizarem, mais pessoas poderão contribuir para o seu

crescimento.

1 Usando a Distribuição Debian

Debian é uma organização exclusivamente de voluntários dedicada ao desenvolvimento de software livre e a promover os ideais da comunidade de Software Livre. Essa organização desenvolve software baseado no kernel do Linux.

Já tive a oportunidade de trabalhar com várias distribuições, mas sem resto de dúvidas o Debian é a distribuição mais organizada de todas, mas tem muito caminho a percorrer, e nós faremos de tudo ao alcance para que esses objetivos sejam atingidos.

1.1 Compreendendo o Processo de Boot

Ao ligar o computador, o primeiro software executado é o BIOS (Basic Input Output System) da placa mãe, que faz a contagem da memória RAM, uma detecção rápida dos dispositivos instalados e por fim executa o BOOT carregando o sistema operacional a partir de uma tecnologia de armazenamento. Este procedimento inicial é chamado de POST (Power-On Self Test), que nada mais é do que um auto-teste para ver se aparentemente está tudo correto.

Durante o BOOT (bootstrap) o BIOS carrega o Sistema Operacional, lendo o primeiro setor do disco rígido o `Master Boot Record` chamado de `MBR`, também conhecido como trilha zero. No MBR vai o gerenciador de `boot` (Boot Manager). Os dois mais usados no Linux são o `lilo` e o `grub`.

Na verdade, no MBR mesmo vai apenas um `bootstrap`, um pequeno software que instrui o BIOS a carregar o executável do `lilo` ou `grub` em um ponto específico do HD e após carregar, executar-lo. Lembre-se que o MBR propriamente dito ocupa um único setor do HD, apenas 512 bytes. Não é possível armazenar muita coisa diretamente nele, mas nada impede que esse pequeno programa de 512 bytes, aponte para qualquer tipo de programa.

O gerenciador de `boot` utiliza os primeiros 446 bytes do `MBR`. Os 66 bytes restantes são usados para armazenar a tabela de partições, que guarda informações sobre onde cada partição começa e termina.

O `lilo` e o `grub` podem ser configurados ainda para carregar o Windows ou outros sistema instalados. Muitas distribuições configuram isso automaticamente durante a instalação.

Quando se opta por dar o `Boot` pelo `Linux`, o programa de `Boot Manager`, carrega um programa chamado de `kernel`. No caso do `Linux`, é um arquivo compactado somente-leitura, geralmente é o arquivo `/boot/vmlinuz`. Ele é descompactado em uma área reservada da memória `RAM`, assim que termina de descompactar o seu conteúdo, o mesmo começa a ser executado.

Este executável principal do `kernel` nunca é alterado durante o uso normal do sistema, ele muda apenas quando você recompila o `kernel` manualmente ou instala uma nova versão.

Depois de carregado, a primeira coisa que o kernel faz é montar a partição raiz, onde o sistema está instalado, inicialmente como somente leitura. Neste estágio ele carrega o `init`, o software que inicia o `boot` normal do sistema, lendo os `scripts` de inicialização e carregando os `módulos` e softwares especificados neles.

1.2 Inicialização do Sistema

O arquivo de configuração do `init` é o `/etc/inittab`. Ele é geralmente o primeiro arquivo de configuração lido durante o `boot`. A principal tarefa dele é carregar os demais `scripts` de inicialização, usados para carregar os demais componentes do sistema.

Dentro do arquivo `/etc/inittab` existe uma chamada para todos os arquivos que iniciam o nome pela letra "*S*" que se encontram no diretório `/etc/rcS.d`. Esta pasta contém scripts que devem ser executados sempre, a cada boot e são responsáveis por etapas **fundamentais** do `boot`. Dentre alguns exemplos de `scripts` e programas que são executados nesta etapa são (encontrados em `/etc/rcS.d` porém com uma letra S):

- `keymap.sh` - Carrega o layout do teclado que será usado no modo texto, já o `KDE` possui um configurador próprio, o `kxkb`, que é configurado dentro do painel de controle. O *layout* usado pelo `kxkb` subscreve o configurado pelo `keymap.sh`
- `checkroot.sh` - Executa o `fsck`, `reiserfsck` ou outro programa adequado para verificar a estrutura da partição raiz (a partição onde o sistema está instalado), corrigindo erros causados por usos incorretos do sistema. Este processo é análogo ao scandisk do Windows.
- `Modutils` – Função de ler os arquivos `/etc/modules` e `/etc/modules.conf`, ativando a placa

de som, rede e todos os outros dispositivos de hardware que não são ativados pelo kernel e sim por adendos ao mesmo (suporte não foi habilitado diretamente no Kernel).

- checkfs.sh - Parecido com o checkroot.sh, ele se destina a checar as demais partições do HD.
- mountall.sh – Monta as partições do disco que estão especificado no arquivo /etc/fstab. Todas montagens de redes ou partições que desejar serem acessível após o boot, você deverá acrescentar no /etc/fstab.
- networking - Ativa a rede, carregando a configuração de IP, DNS, gateway, ou obtendo a configuração via DHCP. A configuração da rede é armazenada dentro do arquivo /etc/network/interfaces.

Depois dos scripts principais, são executados os scripts correspondentes ao runlevel padrão do sistema, que é configurado no /etc/inittab no nosso caso, o padrão é **2.**

O número 5 indica o runlevel que será usado, que pode ser um número de 1 a 5. Cada runlevel corresponde a uma pasta, com um conjunto diferente de scripts de inicialização. É uma forma de ter várias possibilidades para uso do sistema, em diferentes situações. A configuração mais comum é a seguinte:

- Runlevel 1 - Single user. É um modo de recuperação onde nem o modo gráfico, nem o suporte a rede, nem nenhum outro serviço não essencial é carregado, de forma a minimizar a possibilidade de problemas. A idéia é que o sistema inicialize para que você possa corrigir o que está errado. Essa operação é análoga ao Modo de Segurança do Windows.
- Runlevel 3 - Boot em modo texto. Neste modo todos os serviços são carregados, com exceção do gerenciador de gráfico de boot, nesse caso que podem ser KDM, GDM ou XDM.
- Runlevel 5 - É o modo padrão do Debian. Inicialização normal, com modo gráfico e todos os demais serviços.

No caso de runlevel 5, são carregados os scripts de dentro da pasta /etc/rc5.d, enquanto que usando o runlevel 3, seriam carregados os scripts dentro da pasta /etc/rc3.d. Nada impede que você modifique a organização dos arquivos manualmente, de forma a fazer o X carregar também no runlevel 3, ou qualquer outra coisa que quiser.

1.3 Ativando e desativando serviços

Na distribuição Debian os scripts que iniciam os serviços de sistema ficam **todos** dentro da pasta /etc/init.d. Para parar, iniciar ou reiniciar qualquer serviço, use:

```
/etc/init.d/<NomeDoServiço>
```

Onde o nome do serviço, deve ser substituído pelo nome do script desejado. Se você executar como apresentado acima, sem parâmetros, o mesmo apresentará na tela todas opções possíveis sem executar nada. Exemplo:

```
/etc/init.d/apache2
```

Como resposta ao comando incompleto, teremos todas as opções possíveis:

```
Usage: /etc/init.d/apache2 start|stop|restart|reload|force-reload
```

Os scripts que estão na pasta /etc/init.d servem como repositório de scripts para chamar os executáveis dos servidores. Eles apenas fazem as verificações necessárias e em seguida inicializam ou encerram os executáveis propriamente ditos, que em geral estão na pasta /usr/bin. A pasta /etc/init.d contém scripts para quase todos os servidores que estão instalados no sistema. Quando você instala o Samba pelo apt-get por exemplo, é criado o script /etc/init.d/samba para gerenciar o seu funcionamento.

O que determina se o Samba será executado ou não durante o boot não é o script na pasta /etc/init.d, mas sim um link simbólico criado dentro de uma das pastas de inicialização. Por padrão são executados primeiro o que está dentro da pasta /etc/rcS.d, e em seguida o que estiver dentro da pasta /etc/rc5.d.

Os números antes dos nomes dos serviços dentro da pasta /etc/rc5.d determinam a ordem com que

eles vão ser executados. Você vai querer que o `firewall` seja sempre ativado antes do `Samba` por exemplo.

O *"S"* de *start*, indica que o serviço vai ser inicializado. A partir daí o sistema vai inicializando um por vez, começando com os com número mais baixos. Caso dois estejam com o mesmo número, eles são executados em ordem alfabética. Para que um determinado serviço pare de ser inicializado automaticamente no `boot`, basta apagar o arquivo (`script` ou `link simbólico`) dentro da pasta rc5.d.

1.3.1 Inicialização Gráfica

No Linux, o *X é o* servidor gráfico, responsável prover a infraestrutura necessária em operações gráficas, é ele que controla o acesso à placa de vídeo, lê as teclas digitadas no teclado e os click's do mouse, também oferece todos os recursos necessários para os programas criarem janelas e mostrarem conteúdo na tela.

Se você chamar o X sozinho, a partir do modo texto (digitando X), você verá apenas uma tela cinza, com um X que representa o cursor do mouse. Ou seja, o X é apenas uma base, para os gerenciadores de janelas.

Se você chama-lo com o comando `xinit` ou `xinit -- :2` você já abrirá junto uma janela de terminal, que poderá ser usada para abrir programas. Porém ao abrir qualquer programa gráfico você perceberá que algo está estranho. A janela do programa é aberta, mas fica fixa na tela, você não tem como minimizá-la, alternar para outra janela, e outros recursos.

Isto acontece por que estas tarefas são controladas pelo gerenciador de janelas, que não é carregado com o comando `xinit`. Existem vários gerenciadores de janelas, como o `KDE`, `Gnome`, `WindowMaker`, entre outros, sendo que você pode escolher qual lhe mais agrada.

O `Xfree` utiliza uma arquitetura cliente-servidor, onde o *X* em sí atua como o servidor e os programas como clientes, que recebem dele os clicks do mouse e as teclas digitadas no teclado e enviam de volta as janelas a serem mostradas na tela.

A grande vantagem deste sistema é que além de rodar programas localmente é possível rodar programas instalados em outras máquinas da rede. Existem várias formas de fazer isto. Você pode por exemplo abrir uma janela de terminal dentro do X, conectar-se à outra máquina via SSH (com o comando `ssh -X <IP_da_maquina>`) e começar a chamar os programas desejados ou mesmo obter a tela de login da máquina remota e a partir daí carregar um gerenciador de janelas e rodar todos os programas via rede. Neste caso você precisaria configurar a outra máquina para aceitar as conexões via `XDMCP` e inicializar o *X* com o comando `X -query <IP_da_maquina>` no PC cliente.

1.4 Gerenciador de login Gráfico

Antigamente, era muito comum dar `boot` em modo texto e deixar para abrir o *X* manualmente rodando o comando `startx` apenas quando necessário, pois os PC's eram lentos e o *X* demorava pra abrir.

Atualmente o mais comum é usar um gerenciador de login, como o `KDM` (do `KDE`) ou o `GDM` (do `Gnome`). A função do gerenciador de login é carregar o *X*, mostrar uma tela de login gráfica e carregar o `KDE`, `Gnome` ou outro gerenciador de janelas escolhido. Em geral as distribuições que usam o `KDE` como interface padrão usam o `KDM`, enquanto as que usam o `Gnome` preferem o `GDM`.

O gerenciador de login é aberto como um serviço de sistema, da mesma forma que o apache e outros servidores. Você pode parar o `KDM` e assim fechar o modo gráfico usando o comando `/etc/init.d/kdm stop` e reabri-lo a partir do modo texto com o comando `/etc/init.d/kdm start`.

Como sempre, tudo é aberto através de um conjunto de scripts. O `KDM` por exemplo guarda a base das configurações no arquivo `/etc/kde3/kdm/kdmrc` e coloca um conjunto de scripts de inicialização, um para cada interface instalada dentro da pasta `/usr/share/apps/kdm/sessions/`.

A configuração do kdmrc serve para configurar as opções da tela de login, que vão desde opções de aparência, até a opção de aceitar que outras maquinas da rede rodem aplicativos remotamente via `XDMCP`. Ao fazer login, é executado o script correspondente à interface escolhida. Ao usar o `Fluxbox` por exemplo, é executado o script `/usr/share/apps/kdm/sessions/fluxbox`.

Até mesmo o comando `startx` é um `script`, que geralmente vai na pasta `/usr/X11R6/bin/`. Você pode alterá-lo para carregar o que quiser, mas normalmente ele carrega o gerenciador especificado no arquivo .xinitrc, dentro do `home` do usuário ou `/etc/X11/xinit/xinitrc`, que por sua vez pode chamar `/etc/X11/Xsession`.

Atualmente estão em uso no mundo Linux duas versões diferentes do *X*, o `Xfree` e o `Xorg`. O `Xfree`

o projeto mais antigo e tradicional, o grupo que originalmente portou o *X* para o Linux e foi o principal mantenedor do projeto desde então.

Com o passar o tempo, começaram a surgir críticas, principalmente direcionadas à demora para incluir correções e atualizações nos drivers existentes. Isto foi se agravando com o tempo, até que uma decisão dos desenvolvedores em fazer uma pequena mudança na licença em vigor a partir do `Xfree` 4.4 foi a gota d'agua para que um consórcio formado por membros de várias distribuições desenvolvedores descontentes com o modo de desenvolvimento antigo se juntassem para criar um `fork` do `Xfree`, o `X.org`.

O `X.org` utilizou inicialmente a última versão de desenvolvimento da série 4.3 do `Xfree`, disponibilizada antes da mudança da licença. Desde então foram incluídos muitas atualizações e correções, como novos driver's e vários recursos visuais, como por exemplo suporte à janelas transparentes. A página oficial é a `http://x.org`.

Inicialmente as diferenças eram pequenas, mas como o `X.org` tem o apoio das principais distribuições e está sendo desenvolvido num ritmo muito mais rápido, a tendência é que ele substitua inteiramente o `Xfree` num futuro próximo.

Para quem configura, a principal diferença está nos nomes do arquivo de configuração e utilitários. As opções dentro do arquivo continuam as mesmas, incluindo os nomes dos driver's (radeon, nv, intel, sis, etc.) é possível inclusive usar um arquivo de configuração de uma distribuição com o Xfree em outra (instalada na mesma máquina) com o `X.org`. Aqui vai uma pequena tabela com algumas diferenças. Arquivo de configuração principal:

```
/etc/X11/XF86Config-4
/etc/X11/xorg.conf
```

Utilitários de configuração:

```
xf86cfg - xorgfg
xf86config - xorgconfig
```

É possível também eliminar estas diferenças criando um conjunto de links apontando para os nomes trocados. Assim o XF86Config vira um link para o `xorg.conf` por exemplo, fazendo com que usuários desavisados e até utilitários de configuração consigam encontrar os arquivos sem muitos problemas.

1.4.1 Entrar no Linux pelo modo texto

Existem algumas maneiras de desabilitar todos os daemons X que iniciam automaticamente (XDM, GDM, KWM,...)

- execute `update-rc.d ?dm stop 99 1 2 3 4 5 6.`
- insira "exit 0" no início de todos os arquivos `/etc/init.d/?dm`.
- renomeie todos os arquivos `/etc/rc2.d/S99?dm` para `/etc/rc2.d/K99?dm`.
- remova todos os arquivos `/etc/rc2.d/S99?dm`.
- Execute : `/etc/X11/default-display-manager`

Aqui, o número em rc2.d deve corresponder ao nível de execução especificado em `/etc/inittab`. Também, `?dm` significa que você precisa executar o comando várias vezes substituindo-o com todos entre `xdm`, `gdm`, `kdm` e `wdm`.

Somente o primeiro é a verdadeira maneira que deve ser usada no Debian. O último é fácil mas funciona somente no Debian e requer que você configure o display manager novamente depois usando dpkg-reconfigure. Os outros são métodos genéricos para desabilitar daemons.

1.5 Comandos Básicos do Linux

O Sistema Operacional Linux, possui vasta gama de comandos para as mais variadas funções. Se não bastasse isso, pelo fato de ser um software livre, a todo instante, programadores vem desenvolvendo novos aplicativos/utilitários, que não deixam de ser “comandos”. Para não tornar a lista de comandos muito extensa, foram selecionados alguns dos principais comandos. Caso deseje mais informações sobre os comandos a seguir, digite `man <Nome do Comando>`. Ex: `man ls`. Para sair do manual aperte a tecla *“Q” (quit)*.

1.5.1 Comandos básicos para operação com arquivos ou diretórios:

```
ls [-al]: listagem do diretório.
cp [-ir]: copiar arquivos.
mv [-i]: mover ou renomear arquivos.
rm [--]: deletar arquivos.
mkdir/rmdir: cria/deleta diretórios.
ln -s path link: cria links simbólicos (symlinks) para arquivos ou diretórios.
cd : troca de diretório.
pwd: mostra o diretório atual.
```

1.5.2 Outros comandos de aplicações diversas:

```
file: determina o tipo do arquivo (/etc/magic).
cat: exibe o conteúdo do arquivo na tela.
head / tail: exibe linhas no inicio / fim do arquivo.
less / more: lista o conteúdo do arquivo.
man filename: manual online do programa.
ctrl+alt+del/reboot: reinicia o sistema.
shutdown -h now/halt: desliga o computador.
```

1.5.3 Combinações de comandos:

```
CTRL+C: sai (kill) do programa.
CTRL+ALT+BackSpace: sai (kill) do servidor X.
CTRL+L: limpa a tela.
CTRL+A / E: move o cursor para o início / fim da linha.
CTRL+U / K: deleta da posição do cursor até o início / fim da linha.
CTRL+H: deleta palavra anterior ao cursor.
CTRL+R: busca comando digitado no history do bash.
CTRL+D: logout (para isto altere ou unset a var. $IGNOREEOF).
CTRL+Z: coloca a operação atual e suspensão, usar com bg, fg, jobs.
```

1.5.4 Operação com terminais:

```
stty -a: lista configurações do terminal.
reset: reseta o terminal (volta ao normal).
(SHIFT)PGUP/PGDN: barra de rolagem do bash.
TAB: auto-completa os comandos digitados no terminal.
MOUSE2/3: cola o texto selecionado (gpm).
CTRL+S (Scroll Lock): desabilita o vt.
CTRL+Q (Scroll Lock): habilita o vt (tente isto caso o terminal trave).
ALT+Fx: muda de console. CTRL+ALT+Fx: muda de console em modo gráfico.
```

1.5.5 Informações de Usuários:

```
w: informações gerais sobre usuários logados e seus processos.
who: informações dos usuários atuais (do utmp)
last: listagem do histórico de logins (/var/log/wtmp)
lastlog: retorna informações sobre últimos logins.
```

1.5.6 Processos:

```
CTRL+Z: suspende o processo temporariamente, usar com bg, fg, jobs.
top: os processos que consomem mais recursos do sistema.
jobs: lista as tarefas rodando em fore/background.
bg/fg: manda processo para o back/foreground.
nice/renice: altera prioridades.
ps -auxw: lista todos os processos do sistema.
pstree -p: idem.
time: calcula o tempo decorrente do início ao término de um processo.
```

1.5.7 Matando processos:

```
kill: as opções mais comuns são (onde id é o mesmo que PID):
kill -HUP id-do-processo: reinicia processo.
kill -9 id-do-processo: mata processo.
killall processo: mata processo pelo nome.
```

```
killall -HUP processo: reinicia processo pelo nome.
Ps -aux: mostra os processos.
```

1.5.8 Comandos de Análise do Sistema

```
df -h: espaço livre e ocupado nos discos)
du -sh: espaço ocupado pelo diretório e seus subdiretórios
free: status da memória e swap.
vmstat: status da memória virtual (processos, cpu).
lsdev, lspci: listagem do hardware/dispositivos pci.
pnpdump: retorna configuração das placas ISA PnP.
lsmod / rmmod: lista/remove módulos na memória.
procinfo: cat /proc
xdpyinfo: recursos do servidor X.
showrgbq: retorna a database de cores rgb.
xlsfonts: lista as fontes reconhecidas pelo X.
xset m 5/2 1: ajusta a velocidade e acel. do mouse.
```

1.5.9 Aplicações de Rede

```
lsof -n -i:80 :-i4 = ipv4 e -n = sem resolver hostnames.
fuser -v 80/tcp :lista processos que escutam na porta tcp 80 em modo ps-like.
ping: analise de conexão.
netstat: analise de conexão.
```

1.5.10 Pipes e Redirecionamentos.

```
dmesg | less ; ls -l | more
echo "Broadcast Message" | wall
Através de '<' e '>' é possível definir qual será o stdin e o stdout.
dmesg > dmesg.txt ; more < dmesg.txt
ls -l /tmp >> list.txt (concatena)
ls /admin > list.txt 2>erros.txt
ls /admin > list.txt 2>&1 listagem_e_erros.txt
```

1.5.11 Operadores Lógicos

```
&&: 'e' (retorna true se todas as expressões forem verdadeiras)
||: 'ou' (retorna true se uma das expressões forem verdadeiras)
O sinal ';' executará ambas as expressões independente do retorno. Por exemplo:
make ; make install (os comandos serão executados em sequência)
make && make install (executa make install se não der erro no make)
```

1.5.12 Permissões

chmod: Ao listar as informações de um arquivo ou diretório, o formato é o seguinte: drwxrwxrwx. Respectivamente: diretório (d), permissão do dono (read/write/execute), do grupo (read/write/execute) e de outros (read/write/execute). Por exemplo, para transformar um arquivo em executável:

```
chmod +x nome_do_arquivo    (executável para todos)
chmod g+x nome_do_arquivo   (executável para o grupo)
chmod 755                   (executável): -rwxr-xr-x
```

chown: Para alterar o usuário e o grupo de um arquivo ou diretório (use -R para ser recursivo, ou seja, executar para os sub-diretórios):

```
chown root.root /sbin/firewall.sh

chmod 4700                (suid) set user id para programas
                           que precisam rodar com permissão
                           de root: -rws------
```

Para calcular o valor numérico das permissões, basta considerar o valor do executável como 1, de escrita como 2 e de leitura como 4, que seria o equivalente decimal aos bits:

```
rwx = 111 (todos bits ligados) = 2**2 + 2**1 + 2**0 = 7
```

Dessa forma, uma permissão de leitura e escrita (4+2) para o owner, e de leitura apenas para os outros teria o valor 644. Para calcular a umask, que seria a máscara de permissão aplicada na criação de um novo arquivo, basta então subtrair 666 (ou 777 para diretórios) resultando em umask 022.

1.5.13 Como se encontrar no sistema

```
find [path...] -name [nome_do_arquivo]
find . -name slackware.png
find / -name "*.png" -print (arquivos png do dir. atual)
find /home -size +5000k -print (arquivos com mais de 5Mb)
```

1.5.14 Local de um binário:

```
whereis (ou which) [nome_do_arquivo]
which gcc
gcc: /usr/bin/gcc
```

1.5.15 Criar um banco de dados com a localização de arquivos

```
updatedb
Para pesquisar: locate [nome_do_arquivo]
```

1.5.16 Localizar texto em arquivo:

```
grep [param] [texto] [arquivo]
grep -ni man /var/log/packages/grep.tgz (-i=insensitive -n=número da linha)
      (use ' '(aspas simples) no [texto] para procurar palavra exata.)

ls -l | grep '^-..x' (lista executáveis)
ls -l | grep '^d' (lista diretórios - '^' indica a primeira letra da linha)
```

1.5.17 Operações com texto:

```
comm/diff: compara dois arquivos.
ispell: verficador ortográfico (-d br: dicionário em português).
sort: ordena em ordem crescente, alfabética, etc.
uniq: remove linhas duplicadas.
cut: retorna area delimitada (-c5: quinto caracter).
wc: conta linhas, palavras e bytes.
fold: ajusta o texto para a largura especificada.
nl: numera as linhas de um arquivo.
fmt: reformata as linhas de um arquivo.
expand/unexpand: converte tabs em espaços e vice-versa.
tr: remove e substitui caracteres (-d a-d para remover as letras entre a-d, tr a-d
A-D para torná-las maiúsculas).
```

1.5.18 Criando aliases (nomes curtos)

```
alias cdrom.on="mount /dev/hdd /mnt/cdrom"
alias cdrom.off="umount /dev/hdd"
alias zipdisk="mount -t vfat /dev/hdb4 /mnt/zip"
alias rm="rm -i"
alias x="startx -- -nolisten tcp"
```

1.5.19 Utilitários no console

```
whatis/apropos: descrição do programa.
bc: calculadora (ex: echo "scale=2;1/10"|bc //scale são as casas decimais).
nano ou pico: editor de texto simples (nano-editor.org).
jed: editor de texto para programadores.
mc: o midnight commander.
```

1.5.20 Criar um Link Simbólico

Para Criar um Link simbólico, ou seja uma apontador para outro arquivo/diretório, devemos usar:

```
ln [<opções>] origem [<destino>]

 onde as opções mais comuns são:
-s : cria um link simbólico;
-d : link para um diretório;
```

1.6 Gerenciadores de Pacotes

A distribuição Debian é uma das mais organizadas que existem, a única desvantagem para os usuários mais inexperientes, é o fato de lançarem somente uma nova versão estável a cada seis meses (aproximadamente). Para uso em servidores de alto desempenho isso é ótimo, pois tem garantia que a distribuição é realmente estável, no entanto para o usuário doméstico, o fato de somente poder atualizar o seu sistema a cada 6 meses pode parecer uma eternidade (ex: nova versão do KDE). Já o usuário mais avançado não vê problema nisso, pois pode instalar as versão unstable ou testing (não quer dizer que não funcionem bem, somente não passaram pelo tempo de análise da distro, eventualmente pode acontecer que o pacote realmente está com problemas), que o sua Debian estará tão atualizado quanto qualquer outra distribuição.

1.6.1 Usando apt-get e aptitude

O gerenciador de pacotes padrão Debian é o apt (aptitude, apt-get todos comandos são compatíveis entre si). O sistema de empacotamento usa um banco de dados próprio para saber quais pacotes estão instalados, quais não estão e quais estão disponíveis para instalação. O apt-get usa esse banco de dados para saber instalar os pacotes solicitados pelo usuário e também quais pacotes são necessários para que o mesmo rode perfeitamente (Pode-se utilizar o Synaptic, pois o mesmo utiliza o mesmo banco de pacotes que o apt e tem uma funcionalidade muito boa).

- Atualizar[1] banco de pacotes disponíveis: apt-get update/aptitude update
- Instalar[2] pacotes (repositório): apt-get install <NomeDoPacote> <Opções>
  - -h - ajuda
  - -d - baixar arquivos apenas, não instalar
  - -f - conserta erros de instalações de pacotes
  - -s - não agir, apenas simular operação
  - -y - assume *sim* para todas as perguntas
  - -u - mostrar pacotes que serão atualizados além dos especificados
- Reinstalar um pacote: apt-get –reinstall install <NomeDoPacote>
- Remover[3] pacotes instalados: apt-get remove <NomeDoPacote>
- Atualizando[4] os pacotes já instalados: apt-get upgrade
- Atualizando[5] a distribuição instalada: apt-get dist-upgrade

No caso de usar o aptitude (ncurses) sem parâmetros, o mesmo abre uma tela organizada em formato texto, podendo selecionar/de-selecionar, ver o que está instalado e o que pode ser instalado (de acordo com os repositórios selecionados).

Existe uma variação do apt, que é o apt-cdrom. O mesmo pode ser utilizado para instalar pacotes a partir de um cdrom com pacotes. Outra ferramenta apt é o apt-setup que permite modificar de forma automática o arquivo de /etc/apt/sources.list.

Abaixo um exemplo simples de arquivo /etc/apt/sources.list com explicação das seções:

1 Para manter essa lista atualizada, você deve usar o comando apt-get update/aptitude update. Ele procura pelas listas de pacotes nos repositórios indicados no seu arquivo /etc/apt/sources.list. Essa lista pode conter apontamentos para sites http, ftp, drives locais.

2 Caso o pacote não se encontre no cache local (somente estará depois que o pacote estiver instalado), o mesmo será baixado do repositório especificado no /etc/apt/sources.list. O cache local se encontra em /var/cache/apt/archives. Pode-se ainda especificar o tipo de distribuição que você deseja instalar, acrescentando-se / após o nome do pacote e o tipo (testing, unstable, stable), caso não passar nenhuma informação, o padrão é stable.

3 Muitas vezes o pacote atual é dependência de outros pacotes, porém o mesmo informa isso e pede se deseja excluir os outros pacotes que dependem do mesmo (cuidado ao remover uma dependência de um pacote importante). O remove não elimina os arquivos de configuração, para remove-los use apt-get –purge remove <NomeDoPacote>.

4 Compara as versões do banco de dados local com as versões dos repositórios. Para manter os repositórios atualizados, sempre utilize o apt-get update.

5 Essa característica do apt serve para atualizar uma distribuição inteira de uma só vez, seja através da internet ou de um novo CD/DVD.

```
deb http://www.debian.org/debian stable main contrib non-free
deb http://nonus.debian.org/debian-non-US stable non-US
```

Você pode interpretar cada parte da seguinte maneira:

- deb - Identifica um pacote da Debian. A palavra deb-src identifica o código fonte.
- http://www.debian.org/debian - Método de acesso aos arquivos da Debian, site e diretório principal. O caminho pode ser http://, ftp://, file:/.
- stable - Local onde serão procurados arquivos para atualização. Você pode tanto usar o nome de sua distribuição (Woody, Sarge) ou sua classificação (stable, testing ou unstable. Note que unstable é recomendada somente para desenvolvedores, máquinas de testes e se você tem conhecimentos para corrigir problemas. Nunca utilize unstable em ambientes de produção ou servidores críticos, use a stable.
- main contrib non-us - Seções que serão verificadas no site remoto.

1.6.2 Debian Package (dpkg)

Muitas vezes podemos obter um pacote padrão Debian de forma direta (.deb). O dpkg (Debian Package) é o programa responsável pelo gerenciamento de pacotes em sistemas Debian, assim como o rpm é o gerenciador padrão do Red Hat. Sua operação é feita em modo texto e funciona através de comandos. assim caso deseje uma ferramenta mais amigável para a seleção e instalação de pacotes, prefira o dselect (que é um front-end para o dpkg, estilo aptitude) ou o apt.

O dpkg é muito usado por usuários avançados da Debian e desenvolvedores para fins de instalação, manutenção e construção de pacotes.

- Instalar pacotes: dpkg -i <NomeDoPacote>
- Listando pacotes instalados: dpkg -l [<NomeDoPacote>]
- Removendo pacotes instalados: dpkg -r <NomeDoPacote>
- Removendo pacotes juntamente com os arquivos de configuração: dpkg -P <NomeDoPacote>
- Mostrar descrição do pacote: dpkg -I <NomeDoPacote>
- Mostrar pacotes com problemas de instalação: dpkg -C

1.6.3 Alien (Conversor de Pacotes rpm para deb)

O alien é um comando não nativo da Debian, mas é altamente recomendável instalar essa ferramenta para converter pacotes de outras distribuições (LSB, Red Hat, Stampede, Slackware) em pacotes Debian. O procedimento interno é descompactar o pacote do seu formato atual é recriar o mesmo no formato deb, que é reconhecido pelo dpkg.

- Converter pacote: alien <NomeDoPacote>

Atenção: eventualmente, o alien pode se perder, recentemente inutilizei duas distribuições Debian, usando o alien para converter um pacote rpm, o mesmo gerou tantos arquivos temporários que não consegui realizar o boot no computador. É possível reverter essa situação executando um boot assistido. Portanto caso o alien demore muito para converter um pacote, pode estar ocorrendo que o mesmo tenha se perdido. Importante, o alien pode demorar um pouco para converter um pacote.

1.7 Problemas Comuns e Soluções

A distribuição Debian, à principio não possui problemas após a instalação, mas dependendo do hardware que o usuário possui, não é possível auto-detectar tudo, assim, caso aconteça esse tipo de problema, abaixo estão listados as principais soluções adotadas. Outro detalhe é que por *default* a instalação padrão do Kernel não possui certos recursos (suporte a mais de 1Gb de memória RAM, gravação em partições NTFS, entre outras características), necessitando assim, de uma compilação ou recompilação do kernel.

1.7.1 Compilando o Kernel

A compilação do Kernel se torna uma rotina caso você é uma pessoa que adquire constantemente

novos equipamentos, ou então deseja extrair ao máximo o desempenho do seu Linux. Nas próximas sessões serão apresentadas duas formas de compilar o Kernel.

1.7.1.1 Compilação à "Moda Debian"

Compilar o Kernel, ou seja, o núcleo do Linux já era relativamente fácil para quem era da área, porém a Debian, fez com que esse processo se tornasse muito mais fácil. Primeiramente instale os gerenciadores de compilação.

```
apt-get install kernel-package libncurses5-dev
```

Busque na internet, o código fonte do kernel que você desejar (www.kernel.org). Você pode fazer o download com o mecanismo que bem desejar, mas já aproveitando o momento, vou apresentar outra ferramenta. Caso queira, pode usar o `wget <EndereçoArquivo>`, que o mesmo realiza o download para você e disponibiliza o arquivo no diretório atual.

Uma vez realizado o download, copie o arquivo para o diretório `/usr/src` e descompacte o arquivo. Para descompactar o arquivo use `tar -x` e acrescente `z` se o final do arquivo for `gz`, ou `-j` se for `bz2`, seguido das letras `vf`, (`v` é opcional) como por exemplo:

```
tar -xzvf linux-2.6.17.1.tar.gz
```

O comando acima, descompactará o arquivo, mostrando o que está acontecendo (opção `v`), após descompactado, você pode apagar o arquivo compactado. É altamente recomendável que se crie um link simbólico (seção 1.5.20, página 14) para o diretório criado com o nome `linux`. Para isso:

```
ln -s  linux-2.6.17.1 linux
```

Com isso, temos dois diretórios que representam o mesmo conteúdo, esse recurso é importante, pois assim podemos facilmente informar para o sistema buscar os arquivos de dentro do diretório `linux` (link), e quando mudarmos de versão simplesmente modificamos o link. Dessa forma podemos ter mais de uma versão do Kernel, que pode ser usada para questões de compatibilidade e possibilidades de voltar atrás, caso tenhamos problemas com a nova versão.

O próximo passo é compilar o Kernel, mas é sempre bom, partir a compilação de uma estrutura pre-existente, ou seja, um arquivo que informa o que deve ser compilado ou não. Esse arquivo vem por padrão em `/boot/` com o nome `config-2.6.8-2-386` (caso optou pela instalação 2.6 do kernel), para utilizar-lo, faça:

```
cp /boot/config-2.6.8-2-386 /usr/src/linux/.config
```

Com o comando acima, você copiará o arquivo `config-2.6.8-2-386` do diretório `/boot` para dentro `/usr/src/linux/` com o nome de `.config`.

Antes de começar a compilação devemos ajustar o arquivo de configuração para a nova versão do kernel usando algum mecanismo. Inicialmente você deve estar dentro do diretório `/usr/src/linux`. Você pode usar o `make oldconfig`, onde o mesmo carrega o arquivo `.config` e mostra na tela os recursos que não existia no kernel anterior, pedindo se você deseja incluir esse recurso na nova compilação (na dúvida só de um `enter`, mas de uma lida antes), mas não permite modificar as opções que são comuns entre os "kerneis". Ou então usar o `make menuconfig`, onde o mesmo apresenta uma tela, na qual você pode escolher o que deseja ou não, o importante é, assim que entrar no `menuconfig`, usar a setas do teclado para baixo e escolher a opção "`Load an Alternate Configuration File`", para que você use um kernel baseado no arquivo de configuração que já existe (`.config`). Portanto, devemos escolher um desses dois métodos (existem outros como xconfig, gconfig). Usar:

```
make oldconfig
```

ou

```
make menuconfig
```

Agora vamos a compilação propriamente dita, com base no nosso arquivo de compilação, esse

procedimento, gerará um arquivo .deb, com os arquivos necessários para o boot, ou seja, um pacote Debian (seção 1.6 página 15):

```
make-kpkg -initrd kernel-image
```

Após gerar o pacote de instalação, é só instalar o pacote via dpkg (seção 1.6.2 página 16), Importante o pacote será gravado no diretório /usr/src, portanto devemos voltar um nível nos diretórios usando para isso o "cd .." . Após é só instalar o pacote, como abaixo:

```
cd ..
dpkg -i linux-image-2.6.17.1_2.6.17.1-10.00.Custom_i386.deb
```

Pronto, se você usar o lilo ou grub, os mesmo serão ajustados automaticamente. Será acrescentado mais uma opção de boot, sem eliminar as já existentes.

1.7.1.2 Compilação padrão

Antes de mais nada, quero deixar claro que existem outras formas de proceder a compilação, no entanto esse é o método que eu utilizo.

A primeira parte do procedimento é igual para ambos os métodos de compilação, portanto execute os passos da seção 1.7.1.1 da página 17, até a ferramenta que define as opções do novo kernel (menuconfig, oldconfig, xconfig, ...). Você estando no diretório /usr/src/linux, execute os seguintes comandos:

```
make && make modules && make modules_install
```

Caso prefira, pode executar esses três procedimentos um após o outro, no caso acima, foi usado &&, que significa "caso não ocorra erros execute a próxima instrução", essa forma agiliza, pois a compilação é um processo demorado.

Após ter tudo compilado (sem erros), devemos copiar a imagem do kernel para o diretório de boot, nesse caso /boot. Como não informamos o diretório destino da compilação, o mesmo se encontra em /usr/src/linux/arch/i386/boot/ sob o nome de bzImage. É altamente interessante que você coloque um nome de acordo com a versão do kernel no diretório /boot, assim:

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-2.6.17.1
```

No caso acima, estou copiando o kernel recém compilado sob o nome bzImage para o diretório /boot mudando o nome para vmlinuz-2.6.17.1, o nome poderia ser qualquer um, mas como gosto de usar nomes padrão (quando instalei o Debian Sarge, o nome era vmlinuz-2.6.8), coloquei agora esse "vmlinuz", já o -2.6.17.1 é a versão do kernel compilado (vm = virtual memory, linu = linux, z = compactado).

Agora precisamos criar uma imagem inicial para que sejam carregados os módulos do sistema de arquivos. Caso esqueça esse ítem e seu kernel não tem suporte ao seu file system, teremos um possível kernel panic. Para criar essa imagem devemos:

```
mkinitrd -o /boot/initrd.img-2.6.17.1 2.6.17.1
```

No caso acima, você pode dar o nome que bem desejar, mas para seguir o padrão, coloque o nome de initrd.img- versão do kernel, e logo em seguida a versão do kernel, o primeiro número serve para fins visuais (nome do arquivo), já o segundo grava dentro do arquivo a informação da versão. Observe ainda que o arquivo será criado no diretório /boot.

O próximo passo é ajustar o boot manager, nesse caso o grub, se for o lilo, de uma olhada no arquivo de configuração e siga os exemplos. Caso usar o grub, acrescente as seguintes linhas antes das já existentes parecidas com as abaixo no arquivo /boot/grub/menu.lst.

```
title           Debian GNU/Linux, kernel 2.6.17
root            (hd0,2)
kernel          /boot/vmlinuz-2.6.17.1 root=/dev/sda3 ro
initrd          /boot/initrd-2.6.17.1.img
savedefault
boot
```

Você deverá ajustar esse arquivo de acordo com as suas necessidades, sugiro replicar o bloco já pronto e modificar de acordo com as necessidades do seu novo kernel. Observe o exemplo acima, estou usando um HD `Sata` com o Linux na partição 3.

1.7.2 Instalação somente com CD´s

A instalação para quem não tem acesso à internet, no mínimo é tortuoso, e possivelmente desastroso. Para conseguir essa manobra, precisará de no *mínimo* os 4 cd´s iniciais. Usando somente 1, você poderá somente instalar o modo texto do Linux, com o básico para rodar o mesmo. O problema em instalar dessa forma, nem mesmo os upgrades de segurança serão instalados, é possível, mas muito desgastante.

1.7.3 Reconhecimento da Placa de Vídeo e Monitor

Um dos problemas mais comuns durante a instalação da distribuição é o não funcionamento da interface gráfica. Para ajustar esse problema muito comum, devemos instalar dois utilitários e suas dependências:

- `xdebconfigurator`: detecta a placa de vídeo e monitor e armazena em um arquivo temporário;
- `dexconf`: salva os dados detectados no arquivo de inicialização do servidor `X`.

Eventualmente podem ocorrer problemas com algumas placas de vídeo, caso ocorra, execute o procedimento acima e observe a linha `VIDEO DRIVER:` caso apareça `unknown` (desconhecido), você deverá saber qual a placa de vídeo instalada e ajustar o arquivo `/etc/X11/XF86Config-4`. Para isso use um editor e mude a `Section Device`, na opção `Driver`, subtituindo `unknown` pela produtora da placa de vídeo. Ex: `nv` para nvídia, `sis`, para Sys.

1.7.4 Cancelou a instalação pela metade (Instalação `Scratch`)

Uma instalação `Scratch`, não é para qualquer um, no entanto, as vezes você precisa saber realmente o que possui na máquina, assim você pode instalar item-a-item como se fosse do zero. Outro ponto positivo é que você realmente aprenderá a utilizar o Linux se conseguir concluir essa façanha. O objetivo desse tutorial não é explicar como fazer uma instalação desse nível, mas a idéia pode auxiliar a resolver determinados problemas que possam a vir acontecer na instalação normal. Antes de começar a instalação "no osso do peito", sugiro a leitura da seção 1.6, onde são tratados os gerenciadores de pacotes.

A primeira coisa que deverá fazer é ajustar o `locale`, que nada mais é do que as configurações padrões do local onde vivemos (características do país e linguagem). Para isso execute:

```
dpkg-reconfigure locales
```

Marque tudo que tiver `pt-br` (`pt_BR ISO-8859-1`, `pt_BR.UTF8`), usando a barra de espaços, dê um Ok, logo em seguida ele pede qual o `locale` principal, selecione `pt_BR`.

Usando o `apt-setup`, adicione alguns repositórios de `ftp` e `http`, para buscar os pacotes faltantes. Logo após adicionar os novos repositórios, execute um `apt-get update` para atualizar os pacotes dos novos repositórios.

Para instalar o servidor *X*, que é responsável pela execução dos gerenciadores de janelas, devemos executar:

```
apt-get install x-window-system x-window-system-core
```

Após instalado o servidor *X*, devemos instalar um gerenciador de janelas, como por exemplo o KDE, e algumas ferramentas, para isso execute:

```
apt-get install kdebase kdm kde-i18n-ptbr
apt-get install kscreensaver kscreensaver-xsavers rss-glx
apt-get install kmix kuickshow kpdf
apt-get install ark bzip2 unzip unrar zip dosfstools
```

Após tudo instalado, execute o comando abaixo para ativar o ambiente:

```
/etc/init.d/kdm start
```

Com isso já temos uma configuração inicial funcional e executando na interface gráfica, caso desejar instalar outras ferramentas, sugiro a procura na internet sobre os repositórios, bem com, os pacotes necessários.

Outros pacotes interessantes é o OpenOffice, que para instalar use:

```
apt-get install openoffice.org-l10n-pt-br openoffice.org-kde oooqs-kde
apt-get install myspell-pt-br
```

1.8 Configurando as interfaces de Rede

Toda as configurações dos IP´s padrão da distribuição `Debian`, que são ativadas durante o boot, encontram-se em `/etc/network/interfaces`, e sempre que modificar esse arquivo, você pode restartar essa configuração:

```
/etc/init.d/networking restart
```

Um exemplo do /etc/network/interfaces:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
        address 10.1.1.4
        netmask 255.255.255.0
        network 10.1.1.0
        broadcast 10.1.1.255
        gateway 10.1.1.1
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 10.1.1.1
        dns-search rede
```

1.8.1 Criando interfaces Virtuais

Para criar uma interface virtual você deve utilizar o comando `ifconfig`. Esse comando permite configurar todas a possibilidades de uma interface de rede.

Uma interface de rede virtual nada mais é do que uma interface de rede normal com mais de um IP. Exemplos de uso:

```
ifconfig                                        #informações de todas interfaces
ifconfig eth0 192.168.1.2 up                    #ativa a eth0 com o IP
ifconfig eth1 down                              #desativa a eth1
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 #seta IP e máscara para eth0
ifconfig eth0:0 192.168.1.3                     #interface virtual para eth0
ifconfig eth0:1 192.168.1.4                     #outra interface virtual para eth0
```

Use sempre `ifconfig <NomeEth>:<Numero> <configurações normais>`

Uma vez criado uma interface virtual você à utiliza como se tivesse várias interfaces de rede ligadas a um `HUB`. Dependendo do tipo de aplicação essa prática pode significar uma pequena brecha na segurança (fato de redes diferentes no mesmo meio) caso permita acessos ao usuário à esse meio em específico.

Um bom arquivo para se colocar parâmetros de configuração mais avançados durante o `boot`, é o final do arquivo `/etc/init.d/bootmisc.sh`, que sempre é disparado durante o `boot`.

2 Programas Servidores

Um programa servidor é caracterizado por receber conexões através de uma interface de rede, podendo ser física ou lógica. Os programas servidores aqui tratados são baseados na topologia TCP UDP/IP, ou seja, trabalham via `Internet Protocol`. Os protocolos baseados em IP, trabalham com dois valores, onde o primeiro representa o host através do número IP e o segundo valor representa um recurso do host, também conhecida com Porta. Assim, para trabalhar com servidores devemos informar o IP e a Porta do serviço que desejamos.

As portas mais comuns, chamadas de *"portas bem conhecidas" (WELL KNOWN PORT NUMBERS)*, podem ser vistas no próprio regulador de portas IANA:

`http://www.iana.org/assignments/port-numbers`

O conhecimento dessas portas é **vital** para o administrador de redes para saber se tudo está ocorrendo bem, se existem programas mal intencionados, a até mesmo verificar se determinado serviço está ou não rodando.

2.1 Servidor Web Apache

O servidor `Apache` é sem dúvidas o melhor servidor de `web` que existe, no entanto, o mesmo somente consegue expressar seu potencial quando associado a aplicações dinâmicas. No seu estado de instalação, o mesmo somente processa páginas estáticas.

Para que o servidor possa processar uma página em `PHP` ou outra linguagem, entra em ação o módulo apropriado (que deve ser instalado ou ativado), que faz o processamento necessário e devolve ao `Apache` a página no formato `html`. Além do processamento de comandos, para uma aplicação dinâmica, são necessários possibilidades de se acessar uma base de dados. Uma das bases de dados mais utilizadas é o `MySQL`. A combinação de tudo isso forma a solução que é popularmente chamada de `LAMP` (`Linux Apache MySQL PHP`).

2.1.1 Instalando o Apache

Atualmente o `Apache` é distribuído em 2 versões, `Apache 2` e o `Apache 1.3`. A versão 2 traz muitas vantagens, sobretudo do ponto de vista do desempenho, mas por outro lado ele é incompatível com os módulos compilados para o 1.3, e principalmente, muitas opções de configuração são diferentes.

No `Debian` o `Apache 1.3` é instalado através do pacote "**apache**", enquanto o `Apache 2` é instalado através do "**apache2**".

```
apt-get install apache2
```

Depois de instalar os pacotes, inicie/reinicie o serviço com o comando:

```
/etc/init.d/apache2 restart
```

Acessando o endereço `http://127.0.0.1`, você verá uma página de boas-vindas, que indica que o servidor está funcionando. Se não houver nenhum firewall no caminho, ele já estará acessível a partir de outros micros da rede local ou da internet. Por enquanto temos apenas uma versão básica do apache, que simplesmente exibe arquivos `html` colocados dentro da pasta `/var/www`, que por padrão fica sendo o diretório raiz do seu servidor `web`. A página `http://NomeServidor/index.html` é na verdade o conteúdo da pasta `/var/www/apache2-default`. As páginas que você hospedará ficarão armazenadas no diretório `/var/www/<NomeDaPasta>`.

2.1.2 Configuração básica

Ao utilizar o `Apache 2`, o arquivo passa a ser o `/etc/apache2/apache2.conf`. Analisando o conteúdo do arquivo de configuração, vemos que o servidor escuta a porta 80 por default. Basta alterar o 80 pela porta desejada e salvar o arquivo. Para que a alteração entre em vigor, é preciso reiniciar o apache com o comando `/etc/init.d/apache2 restart`.

Você pode também fazer com que o servidor escute em mais de uma porta simultaneamente, usando o recurso Binding. Para isso, basta incluir o parâmetro Listen <Porta> logo abaixo da linha Port 80, Para que ele escute também outras portas, por exemplo, você incluiria as linhas:

```
Port 80
Listen 1080
Listen 2480
```

Caso o servidor tenha mais de uma placa de rede, você pode utilizar o parâmetro Listen <IP_da_placa>:<Porta>. Se, por exemplo, estão instaladas duas placas de rede, uma com o endereço 222.132.65.143 e a segunda no endereço 192.168.0.1 e você quer que ele escute em ambas, nas portas 1080 e 2480, bastaria incluir:

```
Listen 222.132.65.143:1080
Listen 222.132.65.143:2480
Listen 192.168.0.1:1080
Listen 192.168.0.1:2480
```

Não existe limitação para o uso deste recurso. Você pode fazer o servidor escutar quantas portas e placas de rede forem necessárias. Ao utilizar o Apache 2 no Debian, a configuração de portas fica separada, dentro do arquivo /etc/apache2/ports.conf.

2.1.3 Virtual Hosts

O Host Virtual é a possibilidade de hospedar vários sites no mesmo servidor (shared hosting). Neste caso, os arquivos de cada site ficam guardados numa pasta diferente e o servidor se encarrega de direcionar cada visitante ao site correto.

A configuração é quebrada em vários arquivos individuais (um para cada site) armazenados dentro do diretório /etc/apache2/sites-available.

Dentro desse diretório você encontrará o arquivo default que contém uma configuração padrão da página default, que é exibido quando o usuário não especifica um domínio válido no servidor. Para adicionar outros sites, basta criar um arquivo para cada um como abaixo:

```
<VirtualHost *>
            ServerName www.servidor.com.br
            ServerAlias servidor.com.br
            DocumentRoot /var/www/DiretorioComAPagina
</VirtualHost>
```

Caso queira, você pode especificar um IP no local do * (IP-Based). A diretiva ServerAlias (opcional), permite que o site seja acessado tanto com ou sem o www. O DocumentRoot também foi alterado, com relação aos exemplos anteriores para refletir a organização padrão adotada no Apache 2, onde os arquivos são por padrão armazenados dentro da pasta /var/www/<NomeDoDiretório>.

A pasta /etc/apache2/sites-available contém todos os sites disponíveis no servidor. Para que os sites fiquem realmente disponíveis, é necessário criar um link simbólico (seção 1.5.20) para cada um, dentro da pasta /etc/apache2/sites-enabled. São estes links que determinam se o site vai realmente ficar disponível.

2.1.4 Suporte ao PHP

Muitas ISP (Internet Server Provider) insistem em utilizar o Apache 1.3, por já estarem acostumados com os comandos de configuração e por vários motivos já citados (incompatibilidades), não usam o Apache 2, que é muito superior. Assim, serão apresentadas as configurações para ambas versões. Mas se você estiver realizando uma nova instalação, sugiro que use a versão 2.

2.1.4.1 PHP no Apache 1.3

Como comentado na seção 2.1, a instalação padrão do Apache não possui suporte ao PHP, portanto devemos instalar esse suporte manualmente. O suporte a PHP é instalado através do pacote php4 ou php3 (depende da versão). Para instalar use:

```
apt-get install php4
```

Em seguida você deve localizar o módulo `libphp4.so` (ou `libphp3.so`) e ativá-lo na configuração do Apache. A localização padrão do módulo pode variar de distribuição para distribuição, por isso o melhor a fazer é localizá-lo usando os comandos:

```
updatedb
locate libphp4.so
```

No `Debian` o modulo é instalado na pasta `/usr/lib/apache/1.3` (`Apache 1.3`). Para ativá-lo adicione a linha a seguir no arquivo `/etc/apache/httpd.conf`:

```
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
```

É necessário incluir também as linhas que associam as páginas com extensão .php ou .phps com o módulo recém instalado. Portanto adicione também:

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

Após realizar as modificações, salve-as e reinicie o servidor.

```
/etc/init.d/apache restart
```

2.1.4.2 PHP no Apache 2

Geralmente o suporte ao PHP já vem "embutido", na instalação do Apache2, caso não venha, instale o pacote `libapache2-mod-php4` ou `libapache2-mod-php5`:

```
apt-get install libapache2-mod-php4
```

O módulo `libapache2-mod-php4` é instalado dentro da pasta `/usr/lib/apache2/modules/`. Ao invés de adicionar as linhas que ativam o módulo e criam as associações de arquivos no final do arquivo `httpd.conf`, são criados dois arquivos dentro da pasta `/etc/apache2/mods-available/`, com respectivamente a ativação do módulo e as associações de arquivos. Para ativar o suporte a PHP, é preciso copiar ambos para a pasta `/etc/apache2/mods-enabled/`, como exemplificado abaixo:

```
cd /etc/apache2/mods-available/
cp -a php4.conf php4.load ../mods-enabled/
```

Este procedimento de copiar arquivos (ou criar links simbólicos) da pasta `mods-available` para a pasta `mods-enabled` é a forma padrão de ativar módulos diversos no `Apache 2` do `Debian`. Para ativar o suporte a `SSL` por exemplo, copie os arquivos `ssl.conf` e `ssl.load`:

```
cp -a ssl.conf ssl.load ../mods-enabled/
```

A ativação dos módulos pode ser automatizada usando o comando `apache-modconf`, que substitui a necessidade de ficar copiando manualmente os arquivos. Ele pode ser usado tanto em conjunto com o `Apache 1.3`, quanto com o `Apache 2`. Inclua no comando o parâmetro `enable`, seguido do nome do módulo desejado:

```
apache-modconf apache enable mod_php4
```

2.1.4.3 Exemplo para testar o suporte ao PHP

Caso o seu Apache estiver correto e seus modulos de PHP também, crie uma página `html` com o conteúdo abaixo:

```
<html>
   <head>
      <title>Testando PHP</title>
   </head>
   <body>

Exemplo de utilização de variáveis: <br><br>

<?
   $inteiro=10;
   $real=20.0;
   $caracter='V';
?>

A variável $inteiro tem o valor <? echo $inteiro ?>.<br>
A variável $real tem o valor <? echo $real ?>.<br>
O caracter escolhido é o <? echo $caracter ?>.<br>

   </body>
</html>
```

2.1.4.4 Instalando o Suporte à Banco de Dados pelo PHP

Grande parte dos sistemas desenvolvidos em PHP utilizam um banco de dados para gerar as páginas dinâmicas, principalmente o `MySQL` ou `PostgreSQL`. Para utilizá-los, você precisa instalar esses servidores, bem como, os módulos `php4-mysql` (`MySQL`) e `php4-pgsql` (`Postgres`), que permitem aos scripts em `PHP` acessarem o banco de dados. Para isso, devemos instalar esses módulos para o `MySQL` e `Postgres` respectivamente (sempre reiniciar o servidor após modificar os módulos instalados):

```
apt-get install php4-mysql

apt-get install php4-pgsql
```

2.2 “Servidor de Banco de Dados” MySQL

Existe muita discussão se o `MySQL` é ou não um banco de dados, ou se ele é apenas um gerenciador de arquivos de dados com algumas características de banco de dados. Como o enfoque desse material não é a discussão desse mérito, diremos que o `MySQL` é um Gerenciador de Banco de Dados (`GDB`).

O `MySQL` é um banco de dados extremamente versátil, usado para os mais diversos fins. Você pode acessar o mesmo a partir de um script em `PHP`, através de um aplicativo desenvolvido em C ou C++, ou praticamente qualquer outra linguagem. Para instalar o `MySQL`, execute o comando abaixo:

```
apt-get install mysql-server
```

Você deve instalar também os pacotes `mysql-client` (permite acessar o banco) e o `mysql-navigator` (interface gráfica). Antes de iniciar o serviço, rode o comando `mysql_install_db`, para criar a base de dados inicial do `MySQL`, usada para armazenar informações sobre todas as outras criadas posteriormente. E posteriormente ativar o servidor.

```
mysql_install_db
/etc/init.d/mysql start
```

O `MySQL` possui um usuário padrão chamado `root`, que assim como o `root` do sistema, tem acesso completo a todas as bases de dados e é usado para fazer a configuração inicial do sistema, assim como as tarefas de manutenção. Esta conta inicialmente não tem senha, por isso você deve definir uma logo depois de iniciar o serviço, usando o comando `mysqladmin -u root password <NovaSenha>`, incluindo a senha desejada diretamente no comando:

```
mysqladmin -u root password senha123
```

Para testar se o `MySQL` está rodando corretamente, podemos chama-lo:

```
mysql -u root -p
```

2.2.1 Administrador de MySQL e PHP via Web

Muitos programadores PHP quando hospedam uma página no servidor, solicitam uma ferramenta chamada de PHPMyAdmin para facilitar as operações de ajustes e instalação de seus scripts. Para isso instale o pacote phpmyadmin (http://www.phpmyadmin.net/):

```
apt-get install phpmyadmin
```

O phpMyAdmim é um script em PHP, que trabalha em conjunto com o Apache. O script de pós-instalação incluído no pacote do Debian faz a configuração inicial para você, perguntando se ele deve ser configurado para trabalhar em conjunto com o Apache 1.3, com o Apache 2 ou com o Apache-ssl, onde a autenticação e transmissão dos dados são feitos de forma encriptada.

Depois de instalado, acesse o endereço http://127.0.0.1/phpmyadmin/ e você cairá na tela de administração do phpMyAdmin, onde você pode logar-se usando qualquer uma das contas registradas no MySQL. Use o root para tarefas administrativas. Por questões de segurança, a configuração padrão permite que ele seja acessado apenas localmente.

2.3 Servidor FTP

O servidor de FTP mais usado no Linux é o Proftpd, incluído em quase todas as distribuições. Para tarefas onde é necessário ter segurança nas transmissões dos arquivos, é recomendável usar o SFTP, o módulo do SSH que permite transferir arquivos de forma criptografada.

O servidor aceita conexões remotas usando os logins dos usuários cadastrados na máquina. Lembre-se que para adicionar novos usuários você pode usar o comando adduser ou instalar o kuser.

Não é difícil instalar o Proftpd em qualquer distribuição, basta procurar pelo pacote proftpd. Para instalar use:

```
apt-get install proftpd
```

A configuração manual do servidor FTP é feita através do arquivo /etc/proftpd.conf, sempre que modificar as configurações, você deverá reiniciar o servidor /etc/init.d/proftpd restart.

Uma das primeiras opções do arquivo, é a opção Port, que permite alterar a porta usada pelo FTP (padrão é a porta 21).

Em seguida vem a opção MaxInstances, que limita o número de conexões simultâneas ao servidor FTP. Esta opção trabalha em conjunto com a limitação de banda.

Se você quiser limitar o acesso dos usuários à seus diretórios home, adicione a linha DefaultRoot ~ no final do arquivo. No Linux o ~ é um coringa, que é automaticamente substituído pela pasta home do usuário que está logado.

Para ativar a limitação de banda, adicione a linha TransferRate RETR <Taxa>:10, onde a <Taxa> pode ser substituído pela taxa desejada em KB/s.

2.3.1 Criando um FTP Anônimo

A princípio apenas os usuários que tiverem logins válidos no servidor poderão acessar o FTP. Caso você queira abrir um FTP público (Anônimo), adicione estas linhas no arquivo de configuração. Elas ficam comentadas no arquivo original:

```
<Anonymous ~ftp>
 User                    ftp
 Group                   nogroup
 UserAlias               anonymous ftp
 DirFakeUser             on ftp
 DirFakeGroup            on ftp
 RequireValidShell       off
 MaxClients              20
 DisplayLogin            welcome.msg
 DisplayFirstChdir       .message
 <Directory *>
```

```
        <Limit WRITE>
                DenyAll
        </Limit>
 </Directory>
 <Directory incoming>
        Umask  022    022
        <Limit READ WRITE>
                DenyAll
        </Limit>
        <Limit STOR>
                AllowAll
        </Limit>
 </Directory>
</Anonymous>
```

A linha `MaxClients` determina o número máximo de anônimos que poderão se logar no servidor. Esta opção é separada da `Maxclients` principal, que limita o número de usuários com login válido. Você pode permitir um número diferente de usuários válidos dos anônimos.

A opção `DisplayLogin welcome.msg` indica o a mensagem de boas vindas que é mostrada quando os usuários logam no `FTP`. Por padrão é o arquivo `/home/ftp/welcome.msg`.

Os usuários anônimos têm acesso apenas aos arquivos dentro da pasta `/home/ftp`. Este é o diretório raiz para eles, eles não tem como ver muito menos alterar outros arquivos do sistema.

A seção `Directory incoming` mais abaixo cria uma pasta de **upload** (por padrão a `/home/ftp/incoming`) onde os anônimos poderão dar upload de arquivos. A idéia é que você veja periodicamente o conteúdo da pasta e mova o que for útil para a pasta `/home/ftp` para que o arquivo fique disponível para download.

Por padrão os anônimos não podem ver o conteúdo da pasta incoming, podem apenas dar upload. Se necessário, crie o diretório incoming com o primeiro comando, já o segundo informa quem poderá acessar:

```
mkdir /home/ftp/incoming
chown nobody.nogroup /home/ftp/incoming
```

Para acessar o seu servidor, os clientes devem usar o login `anonymous` ou `ftp`, usando um endereço de e-mail qualquer como senha. Uma medida comum ao ativar o upload dos usuários anônimos é usar uma partição separada para o `FTP`, para evitar que algum engraçadinho fique dando upload durante a madrugada até lotar o HD do servidor. Neste caso você precisa apenas adicionar uma linha no arquivo `/etc/fstab`, para que a partição desejada seja montada durante o `boot`.

2.3.2 Criando acesso por nome de usuário com ou sem shell

A forma mais simples de fazer esse acesso, é criar os usuários que terão acesso ao `FTP`, colocando o diretório que terão acesso como seu diretório `home` e bloqueando o uso do `shell`, para que eles não possam acessar o servidor remotamente através de outros meios, via `ssh` por exemplo.

Vamos começar adicionando no arquivo a opção que prende os usuários nos seus diretórios `home` que é `DefaultRoot ~` .Você vai precisar adicionar também a seção para liberar o acesso anônimo ao `ftp`, que vimos acima.

```
<Anonymous ~ftp>
        User                ftp
        Group               nogroup
        UserAlias           anonymous ftp
        DirFakeUser         on ftp
        DirFakeGroup on ftp
        RequireValidShell   off
        MaxClients          20
        DisplayLogin welcome.msg
        DisplayFirstChdir   .message
        <Directory *>
                <Limit WRITE>
                        DenyAll
                </Limit>
        </Directory>
```

```
</Anonymous>
```

O diretório padrão do FTP, onde os visitantes terão acesso aos arquivos é a /home/ftp. Vamos supor que desejo ter três acessos (projeto1, projeto2, projeto3). Primeiro crie os diretórios para cada projeto:

```
mkdir /home/ftp/projeto1
mkdir /home/ftp/projeto2
mkdir /home/ftp/projeto3
```

O próximo passo é adicionar os usuários no sistema, tendo o cuidado de fazer as alterações no diretório home e no shell padrão, para que eles tenham acesso via FTP e apenas à pasta desejada. Para adicionar os usuários, use o comando adduser, como se estivesse criando uma conta normal:

```
adduser projeto1
```

Aparecerá várias informações da inclusão e pedirá a senha, repita esse procedimento para o projeto2 e para o projeto3. Por padrão ele cria a pasta /home/projeto1, que fica sendo o diretório home do usuário criado, porém desejamos que o home seja a pasta /home/ftp/projeto1 onde ele irá dar upload dos arquivos. Para modificar isso, devemos alterar o arquivo /etc/passwd, que é o local onde ficam guardadas as informações dos usuários. Na última linha, você verá:

```
projeto1:x:1005:1005:,,,:/home/projeto1:/bin/bash
```

Devemos alterar o /home/projeto1 para /home/ftp/projeto1 para trocar o home e o /bin/bash para /bin/false para travar o usuário e impedir que ele use o shell no servidor. Se você preferir que além do acesso via ftp, os usuários tenham acesso via ssh, então mantenha o /bin/bash.

Depois das alterações, a linha ficará:

```
projeto1:x:1005:1005:,,,:/home/ftp/projeto1:/bin/false
```

Importante: se você colocar /bin/false como no exemplo acima, você OBRIGATORIAMENTE deverá adicionar o shell /bin/false no arquivo de shell's, para que o mesmo o reconheça como válido. Para isso, edite o arquivo /etc/shells e acrescente /bin/false (Você somente precisa realizar essa tarefa uma única vez).

Como o diretório /home/projeto1, não possui mais utilidade, você pode remover o diretório /home/projeto1, usando rm -rf /home/projeto1/.

Não se esqueça de acertar as permissões da pasta /home/ftp/projeto1 usando chown -R projeto1 /home/ftp/projeto1/.

2.4 Acesso Remoto - Servidor Telnet

O acesso remoto permite utilizar um equipamento remoto como se fosse a maquina local, ou seja, você pode usar todos os comandos conhecidos que os mesmos funcionarão igualmente como se estivesse usando a maquina remota localmente.

Muitos dispositivos como modems ADSL ou pequenos servidores, roteadores, entre outros, possuem várias opções de configuração sem possuir teclado ou monitor embutidos. Nestes casos toda a configuração é feita remotamente, através de algum utilitário de configuração. O mais comum é o uso de alguma interface http, que você acessa de qualquer micro da rede local usando o navegador, ou então o uso do ssh(seção 2.5) ou telnet.

O telnet é um protocolo primitivo que permite rodar comandos remotamente através de uma interface de modo texto. Existem clientes telnet para vários sistemas operacionais, tanto no Linux quanto no Windows, você acessa uma máquina remotamente via telnet usando o comando telnet seguido do endereço IP destino.

O telnet está caindo em desuso por problemas de proteção, sendo substituído pelo ssh, que possui as mesmas funcionalidades porém com muitas vantagens. Para instalar o servidor telnet:

```
apt-get install telnetd
```

O telnet é considerado um comando ultrapassado, no entanto o mesmo é ativado pelo inetd, que é

um script utilizado para chamar vários serviços de uso esporádicos, ou seja, caso não estão sendo solicitados, os mesmos não são carregados para a memória. Para restartar o inetd use /etc/init.d/inetd restart.

2.5 Acesso Remoto - Servidor SSH (Secure Shell)

O ssh é a ferramenta mais utilizada pelos administradores de redes para gerenciar de forma remota estações de trabalho e servidores, de forma ágil e segura, utilizando técnicas criptográficas baseadas em chaves públicas para se comunicar.

Na Debian, o ssh é instalado e executado automaticamente, mas caso seja necessário instalar, utilize:

```
apt-get install ssh
```

Para ativar/desativar o serviço use /etc/init.d/ssh e a função desejada. Pelo padrão Debian, os arquivos de configuração do ssh se encontram em /etc/ssh.

O ssh é dividido em dois módulos. O sshd é o módulo servidor, um serviço que fica residente na máquina que será acessada, enquanto o ssh é o módulo cliente. Para que ele seja inicializado durante o boot, use update-rc.d -f ssh defaults.

A configuração do servidor encontra-se no arquivo /etc/ssh/sshd_config, já a configuração do cliente em /etc/ssh/ssh_config.

2.5.1 Configuração do Cliente

Ao ser habilitado, o padrão do servidor ssh é permitir acesso usando qualquer uma das contas de usuário cadastradas no sistema, pedindo apenas a senha de acesso. Para acessar um servidor digite ssh <login>@<IpServidor>, por exemplo ssh admin@200.248.22.44. Ao acessar micros dentro da rede local, podemos chamá-los pelo nome, ssh admin@flamingo. Neste caso você precisará primeiro editar o arquivo /etc/hosts, incluindo os números de IP das máquinas e os nomes correspondentes. Exemplo:

```
127.0.0.1 localhost
192.168.0.2 server
192.168.0.6 flamingo
```

Além de oferecer acesso via linha de comando (modo textual), o ssh permite rodar aplicativos gráficos remotamente (X11 forwarding). Nestes casos, edite o arquivo /etc/ssh/ssh_config (client ssh) e substitua a linha: ForwardX11 no por ForwardX11 yes.

Outra opção é adicionar o parâmetro -X ao se conectar. O maior problema com o uso de aplicativos remotos gráficos via ssh é que ele só funcionam bem em uma rede local por motivo de banda. O protocolo do X é otimizado para uso local, isso faz com que muitos administradores desabilitem o X11 forwarding no próprio servidor.

Outro problema comum do ssh ocorre quando a conexão é fechada pelo servidor depois de alguns minutos de atividade. Em muitas situações você quer manter a conexão aberta por longos períodos, sem a necessidade de ficar interagindo para que a conexão não caia. Você pode evitar o problema fazendo com que o próprio cliente mande pacotes periodicamente a fim de manter a conexão aberta. Para ativar essa função, adicione a linha ServerAliveInterval 120 no /etc/ssh_config. Dessa forma a cada 120 segundos o ssh cliente manda alguma informação inerte ao servidor, mantendo assim a conexão ativa

2.5.2 Configuração do servidor

Você pode configurar várias opções relacionadas ao servidor ssh, incluindo a porta TCP a ser usada editando o arquivo /etc/ssh/sshd_config.

Dentre as principais opções do arquivo de configuração podemos citar a porta de acesso Port 22 que é a porta padrão usado pelo ssh. Ao mudar a porta do servidor aqui, você deverá usar a opção -p ao conectar a partir dos clientes, para indicar a porta usada. Outra opção é editar o arquivo /etc/ssh/ssh_config (ssh cliente) e alterar a porta padrão usada também por eles.

A opção ListenAddress, que permite limitar o ssh a uma única placa de rede, em casos de micros com duas ou mais placas.

Atualmente utilizamos o protocolo `ssh 2`, mas ainda existem alguns poucos clientes que utilizam a primeira versão do protocolo. Por padrão, o servidor SSH aceita conexões de clientes que utilizam qualquer um dos dois protocolos, o que é indicado na linha `Protocol 2,1`.

A opção `PermitRootLogin yes`, determina se o servidor aceitará que usuários se loguem como `root`. Do ponto de vista da segurança, é melhor deixar esta opção como `no`, pois assim o usuário precisará primeiro se logar usando um login normal e depois virar root usando o `su`.

O `ssh` permite que qualquer usuário cadastrado no sistema logue-se remotamente, mas você pode filtrar os mesmos através da opção `AllowUsers`, que especifica uma lista de usuários que podem usar o SSH. Isso evita que contas com senhas fracas, usadas por usuários que não tem necessidade de acessar o servidor remotamente coloquem a segurança do sistema em risco.

Ao invés de permitir, você pode vetar o uso com a opção `DenyUsers`. Neste caso, todos os usuários cadastrados no sistema podem fazer login, com exceção dos especificados na linha.

A opção `PermitEmptyPasswords no`, faz com que qualquer conta sem senha fique automaticamente desativada no ssh (manter sempre em `no` por questões de segurança).

Para exibir mensagens de advertência antes de login use a opção `Banner = /etc/ssh/banner.txt`, onde banner.txt é o arquivo com a mensagem a ser apresentada.

A opção `X11Forwarding yes` informa se o servidor permitirá que os clientes executem aplicativos gráficos remotamente.

Sempre que modificar alguma das opções, você deverá restartar o servidor com `/etc/init.d/ssh restart`.

2.6 Compartilhamento de Arquivos

Atualmente existem inúmeros protocolos de compartilhamento de arquivos. Os principais são o `NFS` (`Network File System`) e o `SMB` (`Server Message Block`). Ambos funcionam muito bem, cada um com suas vantagens e desvantagens.

2.6.1 NFS – Network File System

O `NFS` é o protocolo padrão de compartilhamento de arquivos do `Linux`. O `NFS` utiliza um outro serviço chamado de `portmap` para gerenciar as requisições dos clientes, este serviço precisa estar ativo para que o `NFS` funcione, ou seja, para inicializar o servidor `NFS` você precisa ativar:

```
/etc/init.d/nfs-kernel-server start
```

A configuração do `NFS` é feita em um único arquivo chamado `/etc/exports`, onde vai a configuração dos diretórios compartilhados. Originalmente este arquivo se encontra vazio.

Para exemplificar o seu uso, vamos compartilhar o diretório`/home/arquivos` como somente leitura, para todos os micros da rede local (classe C 192.168.0.0/24), o diretório `/home/trabalhos` para todas as mesmas máquinas porém com permissão de escrita, e a pasta `/arquivos` somente para a máquina 192.168.0.3 para escrita (conteúdo do arquivo `/etc/exports`). Importante: você não pode deixar espaços entre o IP e o tipo de acesso entre parênteses.

```
/home/arquivos 192.168.0.*(ro)
/home/trabalhos 192.168.0.*(rw)
/arquivos 192.168.0.3(rw)
```

Outra opção, útil em redes locais é a `async`, que permite que o `NFS` transfira arquivos de forma assíncrona, sem precisar esperar pela resposta do cliente a cada pacote enviado. Sem a opção `async` a taxa de transmissão numa rede de 100 megabits fica em geral em torno de 6 a 7 MB/s, enquanto que ao ativá-la, sobre para até 11 MB/s, ficando limitada apenas à velocidade da rede e dos HDs no servidor e cliente. Ao adicionar, a linha de compartilhamento fica `/home/trabalhos 192.168.0.*(rw,async)`.

Você pode usar ainda o parâmetro `noacess`, que permite que você compartilhe apenas os arquivos dentro do diretório, mas não subdiretórios que eventualmente estejam presentes. Depois de incluir todos os diretórios que deseja compartilhar, com suas respectivas permissões de acesso, salve o arquivo e reinicie o serviço para que as alterações tenham efeito:

```
/etc/init.d/nfs-kernel-server restart
```

Embora seja fácil editar diretamente o arquivo /etc/exports, muitas distribuições incluem ferramentas gráficas para gerenciar os compartilhamentos. Caso deseje uma ferramenta gráfica, você pode usar (shares-admin na linha de comando) ou então ->Configuração->Pastas Compartilhadas.

Para montar o compartilhamento manualmente, use (como root) os comandos:

```
mkdir /mnt/arquivos
mount -t nfs 192.168.0.1:/arquivos /mnt/arquivos
```

Sempre que for montar uma unidade, você deve possuir um diretório para representar essa montagem, assim a primeira linha dos comandos acima, cria uma pasta chamada arquivos dentro do diretório mnt. A linha de montagem propriamente dita inclui o sistema de arquivos usado, neste caso o nfs (-t (type) nfs), o endereço IP do servidor, seguido da pasta que ele está compartilhando e, finalmente, a pasta local onde os arquivos ficarão acessíveis.

Ao terminar de acessar o compartilhamento, ou caso precise desligar o servidor, use o comando umount /mnt/arquivos para desmontá-lo. É importante desmontar o compartilhamento antes de desligar o servidor, caso o contrário o cliente continua tentando acessar o compartilhamento sempre que você acessa a pasta onde ele está montado, o que faz com que os gerenciadores de arquivos e outros programas travem ao passar pela pasta, aguardando a resposta do servidor que não está mais lá.

Se você acessa o compartilhamento freqüentemente, pode ganhar tempo inserindo uma entrada referente a ele no arquivo /etc/fstab. Assim você pode montar o compartilhamento usando o comando simplificado, ou configurar o sistema para montá-lo automaticamente durante o boot. Basta incluir a linha no final do arquivo, deixando sempre uma linha em branco após ela. A linha para o compartilhamento que acabamos de montar seria (em /etc/fstab):

```
192.168.0.1:/arquivos /mnt/arquivos nfs noauto,users,exec 0 0
```

Neste exemplo o 192.168.0.1:/arquivos é o IP do servidor, seguido pela pasta compartilhada e o /mnt/arquivos é a diretório local onde este compartilhamento ficará acessível (montado), já o nfs é o sistema de arquivos. O noauto faz com que o compartilhamento não seja montado automaticamente durante o boot. Você pode monta-lo e desmonta-lo conforme for utilizá-lo usando os comandos mount /mnt/arquivos e umount /mnt/arquivos. Uma vez o coamdno no fstab, você precisa especificar apenas a pasta, pois o sistema lê os outros parâmetros a partir da entrada no arquivo.

O parâmetro users permite que você monte e desmonte o compartilhamento usando seu login normal, sem precisar usar o root, já o exec permite executar programas dentro do compartilhamento. Caso você esteja preocupado com a segurança, pode remover as duas opções.

Mais um comando útil ao utilizar o NFS é o showmount -a (root) que mostra uma lista com os diretórios NFS compartilhados na sua máquina que foram acessados e quais máquinas os acessaram desde o último reboot. Não é muito específico, pois não mostra datas nem horários, mas pelo menos permite descobrir se alguém não autorizado está acessando os compartilhamentos.

Por padrão, os compartilhamentos do NFS são montados com a opção hard, isso causa um certo transtorno quando o servidor é desligado ou desconectado da rede, pois os clientes ficam tentando se reconectar ao servidor indefinidamente, fazendo que programas travem ao tentar acessar ou salvar arquivos no compartilhamento e você não consiga desmontá-lo por vias normais até que o servidor volte. Para prevenir este problema, você pode montar os compartilhamentos (nos clientes) usando a opção soft.

Neste caso o compartilhamento é escondido caso o servidor seja desconectado e programas tentando acessa-lo passam a exibir mensagens de *"não é possível ler o arquivo"* ao invés de travarem. Para usar esta opção, adicione a opção -o soft no comando de montagem:

```
mount -t nfs -o soft 192.168.0.1:/home/arquivos /mnt/arquivos
```

Assim a linha no /etc/fstab com a opção ficaria:

```
192.168.0.1:/home/arquivos /mnt/arquivos nfs users,exec,soft 0 0
```

2.6.2 SMB - Server Message Block (Samba)

O Samba é o servidor baseado em blocos de mensagens, que permite compartilhar arquivos e acessar

compartilhamentos em máquinas Windows. Ele é dividido em dois módulos, o servidor Samba propriamente dito e o SmbClient. O SmbClient é responsável por permitir que máquinas Linux usem esse protocolo. Usando o Samba, o servidor Linux se comporta exatamente da mesma forma que uma máquina Windows. Permite compartilhamento de arquivos e impressoras e autenticação de usuários. Você pode configurar o Samba inclusive para tornar-se um controlador de domínio. Atualmente o Samba é mais rápido que o próprio protocolo smb do Windows na tarefa de servidor de arquivos.

Para instalar corretamente o Samba no seu servidor, você precisa dos seguintes pacotes, onde o primeiro instala o servidor Samba, o segundo o cliente de acesso, a documentação e por último o Swat (seção 2.6.2.4):

```
apt-get install samba smbclient samba-doc swat
```

O Swat é uma interface web que auxiliar muito na etapa de configuração, mas opcional, pois você pode tanto editar manualmente o arquivo /etc/samba/smb.conf, quanto usar um arquivo pronto, gerado em outra instalação. Nos clientes que forem apenas acessar compartilhamentos de outras máquinas, instale apenas o smbclient. Depois de instalados os pacotes, use os comandos:

```
/etc/init.d/samba start
```

Por padrão, ao instalar o pacote é criado um link na pasta /etc/rc5.d que ativa o servidor automaticamente durante o boot. Para desativar a inicialização automática, use o comando:

```
update-rc.d -f samba remove
```

Pata ativá-lo novamente depois, use:

```
update-rc.d -f samba defaults
```

O nome do processo do Samba é o nmbd e não samba ou smb.

2.6.2.1 Cadastrando os usuários

Uma vez instalado e rodando o servidor, devemos cadastrar os logins e senhas dos usuários que possuirão acesso ao servidor. O Samba é um *programa a parte*, assim você deverá informar quais usuários vão ter acesso ao seu serviço. Por isso ele só pode dar acesso para usuários que, além de estarem cadastrados no Samba, também estão cadastrados no sistema.

Você pode criar usuários usando o comando adduser ou um utilitário como o user-admin (gksu users-admin ou pelo menu usar Usuários e Grupos, disponível através do pacote gnome-system-tools). Ao usar o adduser, o comando fica:

```
adduser <NomeDoUsuário>
```

Outra opção é criar usuários que terão acesso apenas ao Samba. Esta abordagem é mais segura, pois os usuários não poderão acessar o servidor via ssh ou Telnet, por exemplo, o que abriria brecha para vários tipos de ataques. Neste caso, você cria os usuários adicionando os parâmetros que orientam o adduser a não criar o diretório home e manter a senha desativada até segunda ordem:

```
adduser --disabled-login –no-create-home <NomeDoUsuario>
```

Esse procedimento cria um usuário que pertence ao arquivo de acessos, mas sem diretório de acesso, para todos os fins, existe e pode acessar arquivos do sistema (de acordo com as permissões de acesso), mas, por outro lado, não pode fazer login.

Indiferente da forma como cadastrou os usuários no sistema, você deve cadastrá-los no Samba, usando o comando smbpasswd -a. Todos os usuários cadastrados no Samba se encontram no arquivo /etc/samba/smbpasswd.

```
smbpasswd -a <NomeDoUsuario>
```

Se você mantiver os logins e senhas sincronizados com os usados pelos usuários nos clientes

Windows, o acesso aos compartilhamentos é automático. Caso os logins ou senhas no servidor sejam diferentes, o usuário precisará fazer login ao acessar.

Ao usar clientes Windows 95 ou 98, você deve marcar a opção de login como `Login do Windows` e não como `Cliente para redes Microsoft` (que é o default) na configuração de rede do Windows (Painel de controle, opção Redes).

2.6.2.2 Configurando os Compartilhamentos Manualmente

Toda a configuração do `Samba`, incluindo as configurações gerais do servidor, impressoras e todos os compartilhamentos, é feita num único arquivo de configuração chamado de `/etc/samba/smb.conf`. Programas de configuração, como o `Swat` (seção 2.6.2.4), simplesmente lêem este arquivo e o modificam, por isso é possível usar o `Swat` e a configuração manual.

Estrutura básica de um arquivo de configuração, observe que todas linhas que começam por #, significam que seu conteúdo é comentário:

```
# Esse é um exemplo de arquivo de configuração padrão para o Samba, a primeira parte do
# código se refere a estrutura como um todo, ou seja, do comportamento do servidor como
# um todo. Na segunda parte de  comentários, serão  apresentados  os  compartilhamentos
# propriamente ditos.

[global]
      workgroup = REDE
      netbios name = K8VSE
      server string = %h server (Samba %v)
      name resolve order = lmhosts, host, wins, bcast
      printcap name = lpstat
      encrypt passwords = Yes
      wins support = yes
      preferred master = yes
      domain master = true
      domain logons = yes
      logon path = %Nprofiles%u
      obey pam restrictions = Yes
      passwd program = /usr/bin/passwd %u
      passwd chat=*Enter\snew\sUNIX\spassword:* %n\n *Retype\snew\sUNIX\spassword:* %n\n.
      syslog = 0
      log file = /var/log/samba/log.%m
      max log size = 1000
      os level = 100
      dns proxy = No
      panic action = /usr/share/samba/panic-action %d
      invalid users = root
      printing = cups
      print command = lpr -P %p -o raw %s -r
      lpq command = lpstat -o %p
      lprm command = cancel %p-%j
# include = /etc/samba/dhcp.conf
# client code page = 850
# character set = ISO8859-1
      preserve case = no
      short preserve case = no
      default case = lower

[homes]
      comment = %u's Home Directory
      create mask = 0775
      directory mask = 0775
      browseable = no
      read only = no
      map archive = yes
      writable = yes
      force create mode = 0
      security mask = 0775
      force security mode = 0
```

```
[printers]
      comment = Todas as Impressoras disponíveis
      path = /var/tmp
      create mask = 0700
      guest ok = Yes
      printable = Yes
      browseable = No

[epson]
      comment = Impressora Linux
      security = server
      path = /var/spool/lpd/lp
      printer name = lp
      writable = yes
      public = yes
      printable = yes
      print command = lpr -r -h -P %p %s

# No exemplo acima [epson], informei uma impressora compartilhada, afim de exemplo
# Para criar um compartilhamento, você deve acrescentar um nome entre  colchetes, onde
# esse nome refletirá as características do compartilhamento, idealmente, deve-se usar
# um nome de acordo com o tipo de compartilhamento desejado. Abaixo será  acrescentado
# um exemplo hipotético. Para colocar o exemplo em prática, você  deverá  eliminar  os
# comentários após os :.

#[public]                        : Nome que aparecerá no ambiente de redes;
#     path = /home/publico : Diretório que está sendo compartilhado;
#     available = yes            : Disponibilidade do diretório (yes=sim, no=não);
#     browseable = yes           : Será visível pela rede (idem);
#     writable = yes             : Compartilhamento permite a escrita (idem);
#     valid users = maria, joao  : (Opcional) informa quem pode acessar;
#     write list = maria, joao   : (Opcional) informa quem pode escrever;
#     hosts allow = 192.168.0.2, 192.168.0.5 : (Opcional) Ips permitidos;
```

O arquivo que contém todos os usuários e senhas do `Samba` se encontram em `/etc/samba/smbpasswd`. Sempre que alterar manualmente `/etc/samba/smb.conf`, ou mesmo alterar algumas opções pelo Swat e quiser verificar se as configurações estão corretas, rode o `testparm` (basta chamá-lo num terminal). Ele funciona como uma espécie de `debug`, indicando erros grosseiros no arquivo. Depois de fazer qualquer alteração, reinicie o `Samba` usando o comando `/etc/init.d/samba restart`.

O comando `smbstatus` também é muito útil, pois permite verificar quais estações estão conectadas ao servidor e quais recursos estão sendo acessados no momento.

2.6.2.3 Autenticação Centralizada (PDC)

Para solucionar o problema de sincronização de senhas entre estações, existe a opção de usar o servidor `Samba` como um controlador primário de domínio (PDC), onde ele funciona como um servidor de autenticação para os clientes Windows e (opcionalmente) armazena os perfis de cada usuário, permitindo que eles tenham acesso a seus arquivos e configurações a partir de qualquer máquina onde façam logon.

Ao cadastrar um novo usuário no servidor `Samba`, ele automaticamente pode fazer logon em qualquer uma das estações configuradas. Ao remover ou bloquear uma conta de acesso, o usuário é automaticamente bloqueado em todas as estações. Isto elimina o problema de sincronismo entre as senhas no servidor e nas estações e centraliza a administração de usuários e permissões de acesso no servidor, simplificando bastante seu trabalho de administração.

O primeiro passo é modificar o arquivo de configuração do `Samba`. Existem algumas regras adicionais para transformar o Samba num controlador de domínio. A seção `global` deve conter as linhas `domain master = yes`, `domain logons = yes` e `logon script = netlogon.bat` e **não** deve conter a linha `invalid users = root`, pois precisaremos usar a conta de `root` no `Samba` ao configurar os clientes.

É preciso adicionar também um compartilhamento chamado `netlogon`, que conterá o script de login que será executado pelas estações. Abaixo é apresentado um exemplo de arquivo de configuração do `Samba` para um `controlador de domínio`. Ele não contém as configurações para compartilhamento de impressoras, que você pode adicionar (juntamente com os compartilhamentos desejados) depois de testar a

configuração básica:

```
[global]
     workgroup = REDE
     netbios name = GERENCIADOR
     server string = Controlador de Domínio
     domain master = yes
     preferred master = yes
     local master = yes
     domain logons = yes
     logon script = netlogon.bat
     security = user
     encrypt passwords = yes
     os level = 100

[netlogon]
     comment = Servico de Logon
     path = /var/samba/netlogon
     guest ok = Yes
     browseable = No

[homes]
     comment = Diretorio Home
     valid users = %S
     guest ok = Yes
     browseable = No
```

Ao rodar o comando `testparm`, pois ele verifica a sintaxe e indica erros de configuração. Ao configurar o Samba como `PDC`, ele deve exibir a mensagem `Server role: ROLE_DOMAIN_PDC`.

Cuidado para cadastrar o usuário `root` com a mesma senha no `Samba`. Caso necessário, use o comando `smbpasswd -a root` para cadastrar o root. Crie a pasta `/var/samba/netlogon` e configure corretamente as permissões:

```
mkdir -p /var/samba/netlogon
chmod 775 /var/samba/netlogon
```

Com o `775` estamos permitindo que além do `root`, outros usuários que você adicionar no grupo possam alterar o conteúdo da pasta. Isso pode ser útil caso existam outros administradores de rede além de você. Cadastre agora os logins dos usuários, com as senhas que eles utilizarão para fazer logon a partir das máquinas Windows. Neste caso não é preciso se preocupar em manter as senhas em sincronismo entre o servidor e as estações. Na verdade, as contas que criamos aqui não precisam sequer existir nas estações, pois o login será feito no servidor. Para adicionar usuários novos use:

```
adduser <NomeDoUsuário>
smbpasswd -a <NomeDoUsuario>
```

É importante criar também o diretório `profile.pds` dentro do diretório `home` do usuário, onde o cliente Windows armazena as informações da seção cada vez que o usuário faz logon no domínio

```
mkdir /home/<NomeDoUsuário>/profile.pds
```

Além das contas para cada usuário, é preciso cadastrar também uma conta (sem senha) para cada máquina. Você deve usar aqui os mesmos nomes usados na configuração de rede em cada cliente. Se a máquina se chama *athenas* por exemplo, é preciso criar um login de máquina com o mesmo nome:

```
useradd -d /dev/null -s /bin/false athenas$
passwd -l athenas$
smbpasswd -a -m athenas
```

O $ depois do nome, que indica que estamos criando um login de **máquina**, que não tem diretório home *(-d /dev/null)*, não possui um shell válido *(-s /bin/false)* e está travada *(passwd -l)*. Esta conta é válida apenas no `Samba`, onde é cadastrada com a opção `-m` (*machine*).

Por último, é necessário criar o arquivo `/var/samba/netlogon/netlogon.bat`, um script que é

lido e executado pelos clientes ao fazer logon. Como ele é um arquivo de lote, você pode realizar todos os procedimentos que desejar, como por exemplo:

```
net use H: /HOME
net use x: \\gerenciador\arquivos /yes
```

Este script faz com que a pasta `home` de cada usuário seja automaticamente mapeada como a unidade ***H:*** no cliente, o que pode ser bastante útil para backups por exemplo. Naturalmente, cada usuário tem acesso apenas a seu próprio *home*.

A segunda linha é um exemplo de como fazer com que determinados compartilhamentos do servidor sejam mapeados no cliente. O `net use x: \\gerenciador\arquivos /yes` faz com que o compartilhamento de arquivos seja mapeado como o drive ***X:*** nos clientes.

Mais um detalhe importante é que o arquivo do script de logon deve usar quebras de linhas no padrão MS-DOS e não no padrão Unix. Você pode criá-lo usando um editor de texto do Windows ou usar algum editor do Linux que ofereça esta opção. No Kwrite por exemplo, a opção está em: `Configurar > Configurar Editor > Abrir/Salvar > Fim de linha > DOS/Windows`.

Mais uma configuração útil (porém opcional) é fazer com que o servidor armazene os arquivos e configurações do usuário (recurso chamado `Roaming Profiles`), fornecendo-os à estação no momento em que o usuário faz logon. Isto permite que o usuário possa trabalhar em outras máquinas da rede e faz com que seus arquivos de trabalho sejam armazenados no servidor, diminuindo a possibilidade de perda de dados.

Por outro lado, isto faz com que seja consumido mais espaço de armazenamento do servidor e aumenta o tráfego da rede, já que os arquivos precisam ser transferidos para a estação a cada logon. Isto pode tornar-se um problema caso os usuários da rede tenham o hábito de salvar muitos arquivos grandes na área de trabalho.

Note que o servidor não armazena todos os arquivos do usuário, apenas as configurações dos aplicativos, entradas do menu iniciar, cookies, bookmarks e arquivos temporários do IE e o conteúdo das pastas Desktop, Modelos e Meus Documentos. Para ativar o suporte no `Samba`, adicione as duas linhas abaixo no final da seção `global` do `/etc/samba/smb.conf` (abaixo da linha `logon script = netlogon.bat`):

```
logon home = \\%L\%U\.profiles
logon path = \\%L\profiles\%U
```

A variável *%L* neste caso indica o nome do servidor e o *%U* o nome do usuário que está fazendo logon. Adicione também um novo compartilhamento, adicionando as linhas abaixo no final do arquivo:

```
[profiles]
     path = /var/profiles
     writeable = Yes
     browseable = No
     create mask = 0600
     directory mask = 0700
```

Crie a pasta `/var/profiles`, com permissão de escrita para todos os usuários:

```
mkdir /var/profiles
chmod 1777 /var/profiles
```

Cada usuário passa a ter uma pasta pessoal dentro da pasta (`/var/profiles/<NomeDoUsuário>`) onde as configurações são salvas. Apesar das permissões locais da pasta permitirem que qualquer usuário a acesse, o `Samba` se encarrega de permitir que cada usuário remoto tenha acesso apenas ao seu próprio profile. As estações Windows 2000 e Windows XP utilizam os perfis móveis automaticamente, quando o recurso está disponível no servidor `Samba`. Você pode verificar a configuração e, caso desejado, desativar o uso do perfil móvel no cliente no *Meu Computador > Propriedades > Perfis de Usuário > Alterar tipo*.

Neste ponto a configuração do servidor `Samba` está pronta. Falta apenas configurar os clientes Windows para efetuarem logon no domínio. Nem todas as versões do Windows suportam este recurso. Como controladores de domínio são usados principalmente em redes de médio ou grande porte em empresas, a Microsoft não inclui suporte no Windows XP Home e no XP Starter, de forma a pressionar as empresas a

comprarem o XP Professional, que é mais caro. A configuração muda de acordo com a versão do Windows:

- Windows 2000: acesse o *Meu Computador* > *Propriedades* > *Identificação de rede* > *Propriedades,* coloque aqui o nome do computador (que precisa ser um dos logins de máquinas adicionados na configuração do `Samba`) e o nome do Domínio, que é definido na opção `workgroup =` do `/etc/samba/smb.conf`. Para ter acesso a esta opção você deve estar logado como administrador. Na tela de identificação que será aberta a seguir, logue-se como root, com a senha definida no `Samba`. É normal que a conexão inicial demore dois ou três minutos. Se tudo der certo, você é saudado com uma mensagem *Bem-vindo ao domínio <NomeDoDominio>*. É necessário identificar-se como root ao fazer a configuração inicial, para que seja criada a relação de confiança entre o servidor e o cliente. A partir daí aparece a opção opção *Efetuar logon em: <NomeDoDominio>* na tela de login, permitindo que o usuário faça logon usando qualquer uma das contas cadastradas no servidor. Continua disponível também a opção de fazer um login local.
- Windows 98 ou ME: logue-se na rede (na tela de login aberta na inicialização) com o mesmo usuário e senha que será usado para fazer logon no domínio. Acesse agora o *Painel de Controle* > *Redes* > *Cliente para redes Microsoft* > *Propriedades*. Marque a opção *Efetuar Logon num domínio NT*, informe o nome do domínio e marque a opção *Efetuar logon e restaurar conexões*. Ao terminar, é preciso fornecer o CD de instalação e reiniciar a máquina. Note que as máquinas com o Windows 98/ME não são compatíveis com todos os recursos do domínio, elas acessam o domínio dentro de uma espécie de modo de compatibilidade, onde podem acessar os compartilhamentos, mas não têm acesso ao recurso de perfis móveis, por exemplo.
- Windows XP Professional: o procedimento varia de acordo com a versão do `Samba` usada. Se você está usando uma versão recente do Samba, da versão 3.0 em diante, a configuração é bem mais simples, basta seguir os mesmos passos da configuração no Windows 2000. Comece copiando o arquivo `/usr/share/doc/samba-doc/registry/WinXP_SignOrSeal.reg` (do servidor), que fica disponível ao instalar o pacote *samba-doc*. Esta é uma chave de registro que precisa ser instalada no cliente. Acesse agora as propriedades do *Meu Computador* e na aba *Nome do Computador* clique no botão *ID de rede*. Será aberto um Wizard que coleta o nome do domínio, nome da máquina e login de usuário. Lembre-se que é necessário efetuar o primeiro logon como root. Se não der certo da primeira vez, acesse o *Painel de controle* > *Ferramentas administrativas* > *Diretiva de segurança local* > *Diretivas locais* > *Opções de segurança* e desative as seguintes opções:
  - Membro do domínio: criptografar ou assinar digitalmente os dados de canal seguro (sempre)
  - Membro do domínio: desativar alterações de senha de conta da máquina
  - Membro do domínio: requer uma chave de seção de alta segurança (Windows 2000 ou posterior)

Para confirmar se os clientes estão realmente efetuando logon no servidor, use o comando `smbstatus`.

2.6.2.4 Configurando os Compartilhamentos usando o Swat

O `Samba` pode ser configurado através do `Swat`, que é um utilitário de configuração via web.Manter o `X` instalado e ativo num servidor dedicado é considerado um desperdício de recursos, por isso os desenvolvedores de utilitários de configuração evitam depender de bibliotecas gráficas. Deste modo, mesmo distribuições minimalistas podem incluí-los.

O `Swat` é inicializado através do `Inetd`. A função do `inetd` e `xinetd` é parecida, eles monitoram determinadas portas TCP e carregam serviços sob demanda. Isto evita que utilitários que são acessados esporadicamente (como o Swat) precisem ficar ativos o tempo todo, consumindo recursos do sistema. Apesar disso, a configuração dos dois é diferente: no caso das distribuições que usam o inetd, você ainda precisa adicionar (ou descomentar) a linha abaixo no arquivo de configuração do `inetd`, o `/etc/inetd.conf`:

```
swat stream tcp nowait.400 root /usr/sbin/tcpd /usr/sbin/swat
```

Para que a alteração entre em vigor, reinicie o inetd com o comando:

```
/etc/init.d/inetd restart
```

Para acessar o `Swat`, basta um browser disponível e acessar o endereço *http://localhost:901*. No prompt de login, forneça a senha de root (do sistema) para acessar. Ao abrir o `Swat`, na parte de cima estão

os links para as seções da configuração.

Na seção `Password`, você pode cadastrar usuários, substituindo o uso manual do comando `smbadduser -a`. Neste caso, você precisará primeiro cadastrar os usuários utilizando comando `adduser`, o `Swat` apenas cadastra os usuários no `Samba`.

Em seguida, acesse a seção `Globals`, que engloba todas as configurações de rede e acesso. Nas opções `workgroup` e `netbios name`, você deve colocar o nome do computador e o grupo de trabalho a que ele pertence, como faria numa máquina Windows.

A opção `netbios aliases` permite criar *apelidos* para o servidor, de modo de que ele possa ser acessado por mais de um nome. Usando um alias, o servidor realmente aparece duas vezes no ambiente de rede, como se fossem duas máquinas.

A próxima opção é a `interfaces`, que permite limitar os acessos ao servidor se você tiver mais de uma placa de rede.

Na seção `Security Options` chegamos a uma das decisões mais importantes, decidir entre entre utilizar segurança com base no login do usuário (`user`) ou com base no compartilhamento (`share`).

A opção *share* oferece um nível de segurança semelhante ao de uma máquina Windows 98. Os compartilhamentos podem ser acessados por todos os usuários, através da conta guest. Em compensação, esta opção é a mais simples de configurar e pode ser útil em pequenas redes onde não há necessidade de segurança.

A opção *user* é a mais recomendável, pois permite especificar exatamente quais usuários terão acesso a cada compartilhamento, como num servidor NT ou Windows 2000. Naturalmente, para que isso funcione, é necessário que você tenha registrado todos os usuários no Linux e no Samba (como vimos anteriormente), e que os clientes Windows efetuem login na rede usando estes mesmos logins e senhas, ou os forneçam na hora de acessar os compartilhamentos.

Escolhendo este modo, as permissões de acesso aos compartilhamentos do `samba` ficam condicionadas às permissões de acesso de cada usuário. Por exemplo, se você compartilhar a pasta `/home/<NomeDoUsuario>/arquivos`, por default apenas o usuário especificado terá permissão para gravar novos arquivos e alterar o conteúdo da pasta. Para que outros usuários tenham acesso à pasta, você deve dar permissão a eles, criando um novo grupo e dando permissão de escrita para os integrantes do grupo ou adicionando os demais usuários no grupo do usuário, e configurando as permissões de acesso de forma que o grupo possa escrever na pasta.

Mais abaixo, temos a opção `Encrypt Password`. Ela também é importantíssima, e deve ser configurada de acordo com a versão do Windows que rodar nas máquinas clientes. O Windows 95 original **não** suporta encriptação de senhas, por isso só poderá se conectar ao servidor caso a opção seja configurada com o valor "No". Porém, o Windows 95 OSR/2, Windows 98/SE/ME, Windows NT, Windows 2000 e Windows XP utilizam senhas encriptadas, então, ao utilizar máquinas com qualquer um destes sistemas (que é o mais provável) a opção deve ser configurada como *Yes*, caso contrário o `Samba` simplesmente não conseguirá conversar com as máquinas Windows.

A partir do Samba 3 existe a opção de fazer com que o próprio `Samba` mantenha as senhas dos usuários sincronizadas em relação às senhas dos mesmos no sistema. Para ativar este recurso, ative a opção `unix password sync` no Swat. Originalmente, esta opção fica desativada e aparece apenas dentro das opções avançadas. Para chegar até ela você deve clicar no botão `Change View To: Advanced` no topo da tela. Depois de alterar, clique no `Commit Changes`.

Para que tudo funcione, é necessário que as opções `passwd program` e `passwd chat` estejam configuradas com os valores: `/usr/bin/passwd %u` e `*Enter\snew\sUNIX\spassword:* %n\n *Retype\snew\sUNIX\spassword:* %n\n .` . Estes já são os valores padrão no `Swat`, mas não custa verificar.

A opção `Hosts Allow` deve incluir os endereços IP de todos os computadores que terão permissão para acessar o servidor. A opção `Hosts Deny` por sua vez permite especificar máquinas que não terão permissão para acessar o servidor. Você pode usar o `Hosts Deny` para estabelecer exceções ao dito na opção `Hosts Allow`.

Na seção `Browse Options`, a opção `OS Level` permite especificar qual chance o servidor Linux terá de ser o `Master Browser` do grupo de trabalho ou domínio. Sempre que você estiver configurando o Samba para ser o servidor principal, é desejável que ele seja o `master browser`.

Para isso, configure esta opção com um valor alto, 100 por exemplo, para que ele sempre ganhe as eleições. O default dessa opção é 20, que faz com que ele perca para qualquer máquina Windows NT,

Windows 2000 ou Windows XP. Para completar, deixe a opção `Local Master` e `Preferred Master` como *Yes*.

Abaixo, deixe a opção Wins Support ativada (Yes). A opção Wins Server deve ser deixada em branco, a menos que exista na rede algum servidor Wins (rodando o NT server ou o 2K server) ao qual o servidor Linux esteja subordinado. Caso o único servidor seja a máquina Linux, você pode configurar as máquinas Windows para utilizá-la como servidor Wins, para isto basta colocar o seu endereço IP no campo *Servidor Wins* na configuração de rede das estações.

Terminando, pressione o botão `Commit Changes` no topo da tela para que as alterações sejam salvas no arquivo `/etc/samba/smb.conf`.

Uma observação importante é que o `Swat` lê o arquivo `/etc/smb/smb.conf` ao ser aberto, lendo as opções configuradas e mostrando-as na interface, mas gera um novo arquivo sempre que você clicar no `Commit Changes`. Ao ler o arquivo, ele procura por trechos específicos de texto, ignorando tudo que for diferente. Isso faz com que ele remova qualquer tipo de comentário incluído manualmente no arquivo.

Depois de cadastrar os usuários no sistema e no `Samba` e configurar a seção `Globals`, falta apenas configurar as pastas que serão compartilhadas com as estações, através da seção `Shares`.

Cada usuário válido cadastrado no sistema possui automaticamente um diretório home. Estas pastas ficam dentro do diretório /home e podem ser usadas para guardar arquivos pessoais, já que, a menos que seja estabelecido o contrário, um usuário não terá acesso à pasta pessoal do outro. Além dos diretórios home, você pode compartilhar mais pastas de uso geral.

Para criar um compartilhamento, basta escrever seu nome no campo no topo da tela e clicar no botão `Create Share`.

Depois de criado um compartilhamento, escolha-o na lista e clique no botão `Choose Share` para configurá-la. Você verá uma lista de opções, contendo campos para especificar usuários válidos e inválidos, usuários que podem ou não escrever no compartilhamento, nomes ou endereços de máquinas entre outras opções.

O campo `path` é o mais importante, pois indica justamente qual pasta do sistema será compartilhada. O nome do compartilhamento diz apenas com que nome ele aparecerá no ambiente de rede, que não precisa necessariamente ser o mesmo nome da pasta.

A opção `comment` permite que você escreva um breve comentário sobre a pasta que também poderá ser visualizado pelos usuários no ambiente de rede. Este comentário é apenas para orientação, não tem efeito algum sobre o compartilhamento.

A opção `read only` determina se a pasta ficará disponível apenas para leitura ou se os usuários poderão também gravar arquivos. Você pode também determinar quais máquinas terão acesso ao compartilhamento através das opções `Hosts Allow` e `Hosts Deny`. As configurações feitas aqui subscrevem as feitas na seção `global`.

A opção `browseable` permite configurar se o compartilhamento aparecerá entre os outros compartilhamentos do servidor no ambiente de rede, ou se será um compartilhamento oculto, que poderá ser acessado apenas por quem souber que ele existe. Isso tem uma função semelhante a colocar um "$" numa pasta compartilhada no Windows 98. Ela fica compartilhada, mas não aparece no ambiente de rede. Apenas usuários que saibam que o compartilhamento existe conseguirão acessá-lo. Esta opção tem efeito apenas sobre os clientes Windows, pois no Linux a maior parte dos programas clientes (como o Smb4k) mostram os compartilhamentos ocultos por padrão.

Finalmente, a opção `available` especifica se o compartilhamento está ativado ou não. Você pode desativar temporariamente um compartilhamento configurando esta opção como *No*. Fazendo isso, ele continuará no sistema e você poderá torná-lo disponível quando quiser, alterando a opção para *Yes*.

Um detalhe importante é que os usuários só terão permissão para acessar pastas que o login permite acessar. Por exemplo, no Linux o único usuário que pode acessar a pasta /root é o próprio root, ou outro autorizado por ele. Mesmo que você compartilhe a pasta root através do `Samba`, os demais usuários não poderão acessá-la.

Para editar as permissões de uma pasta, basta abrir o gerenciador de arquivos e, nas propriedades da pasta, acessar a guia `Permissões`. As permissões podem ser dadas apenas ao usuário, para todos os usuários pertencentes ao grupo do usuário dono da pasta, ou para todos os usuários. A opção *Aplicar mudanças a todas as subpastas e seus conteúdos* deve ficar marcada para que as permissões sejam aplicadas também às subpastas.

Terminadas as configurações, o servidor já irá aparecer no ambiente de rede, como se fosse um servidor Windows. Os compartilhamentos podem ser acessados de acordo com as permissões que tiverem

sido configuradas, mapeados como unidades de rede, entre outros recursos.

Para compartilhar uma impressora já instalada na máquina Linux, o procedimento é o mesmo. Dentro do `Swat`, acesse a seção `printers`, escolha a impressora a ser compartilhada (a lista mostrará todas as instaladas no sistema), configure a opção `available` como "yes" e configure as permissões de acesso como vimos anteriormente.

2.6.2.5 Permitindo que os usuários compartilhem pastas

Caso você queira permitir que os usuários também criem compartilhamentos, assim como no Windows, o KDE possui um módulo que resolve essas necessidades, permitindo que os usuários compartilhem arquivos dentro dos seus respectivos diretórios de usuário.

Para que este recurso funcione, você deve instalar o módulo de compartilhamento de arquivos do Konqueror. No Debian, ele é fornecido pelo pacote `kdenetwork-filesharing` (geralmente já vem instalado juntamente com o konkeror):

```
apt-get install  kdenetwork-filesharing
```

Como os usuários podem apenas compartilhar seus próprios arquivos, a possibilidade de danos ao sistema é pequena. Dentro do Centro de Controle do KDE, acesse a seção *Internet & Rede-> Compartilhamento de arquivos*. Clique no Modo administrador, forneça a senha de root e marque a opção Compartilhamento simples.

No botão *Usuários permitidos* você tem a opção de autorizar todos os usuários (permitir a todos os usuários compartilhar pastas) ou autorizar apenas os usuários de um determinado grupo. Neste caso, use o "users-admin" ou outro programa de configuração de usuários e grupos para criar um novo grupo e adicionar os usuários desejados a ele.

Este compartilhamento do KDE faz na verdade um duplo compartilhamento. Além do `Samba`, os compartilhamentos ficam disponíveis na rede através do `NFS`, permitindo que você possa escolher qual protocolo prefere usar em cada caso. Lembre-se de que se, você não quiser o compartilhamento via `NFS`, basta desativar o serviço `NFS`. Para que o compartilhamento funcione, você deverá ter o servidor e o cliente `Samba` instalados no sistema e manter o serviço `SMB` ativo.

2.6.2.6 Acessando máquinas Windows

Importante: O Linux somente somente sabe o que é um IP, caso queira realizar um compartilhamento pelo nome da máquina, você deverá acrescentar o nome de todas as máquinas, bem como o seu IP no arquivo `/etc/hosts`.

É possível se acessar, a partir de uma máquina `Linux` rodando `Samba`, diretórios compartilhados residindo em PC rodando Windows. Para montar o compartilhamento, use o comando `smbmount`, mas não esqueça de compartilhar primeiramente o recurso no Windows e depois montá-lo no Linux. Exemplo:

```
smbmount //K6II500/C /mnt/arqs -o username=Juca, password=tada

df

Sist. Arq.           1K-blocos      Usad Dispon.   Uso% Montado em
/dev/sda3             18263556  10380536   6955268  60% /
tmpfs                   256880         0    256880   0% /dev/shm
tmpfs                    10240       116     10124   2% /dev
//K6II500/C            5002368   4393856    608512  88% /mnt/arqs
```

A maquina Windows chama-se K6II500 e o nome compartilhamento é C. O compartilhamento é montado em /mnt/arqs. Caso você não especificar o usuário e a senha o comando solicitará essas informações.

2.6.3 Feramentas Gráfica - LinNeighborhood

Como o nome sugere, o `LinNeighborhood` visa simular um *ambiente de rede* nos clientes Linux. Ele pode ser usado tanto para acessar compartilhamentos de máquinas Windows quanto de outros micros Linux que estejam rodando um servidor `Samba`. Para instalar no seu sistema use:

```
apt-get install linneighborhood
```

Por default o `linneighborhood` montará todos os compartilhamentos dentro da pasta *mnt*, no seu diretório de usuário, mas você pode montar em outra pasta qualquer se desejar.

Numa máquina que é usada por vários usuários, você pode criar uma pasta *Ambiente de Rede* no diretório raiz e montar todos os compartilhamentos de rede dentro dela. Assim a mesma configuração serve para todos os usuários e você ainda cria um ambiente semelhante ao que eles estão acostumados no Windows.

2.6.4 Outro exemplo de configuração do Samba

Abaixo é apresentado um exemplo completo e comentado da configuração de um servidor samba.

```
# Exemplo de arquivo de configuração para o Samba
# smb.conf,v 1.2.4.6 2002/03/13 18:56:16
# Autor: Roberson Carlos Fox

#================ Global Settings ====================
[global]
# Executa uma ação quando o Samba trava:
panic action = /usr/share/samba/panic-action %d

# Nome do Grupo de Trabalho
workgroup = genova

# Descrição ou comentário
server string = %h server (Samba %v)

# Descomente a linha abaixo caso queira carregar automaticamente as impressoras
; load printers = yes

# Descomente a linha abaixo caso você queira admitir o acesso de usuários usando a
conta de convidados
; guest account = nobody
invalid users = root

# Isto diz ao Samba para usar um arquivo de log para cada máquina que a ele se
conectar
log file = /var/log/samba/log.%m

# Define o tamanho dos arquivos de Log
max log size = 1000

# É uma boa idéia deixar a opção security = user, isto vai obrigar o usuário a
# ter uma conta no servidor, pois sempre que alguém se logar a ele será
# requisitada a senha e o login
; security = user

# Você pode querer usar encriptação de senha para dificultar ainda mais
# a ação de pessoas mal intencionadas deixando como True a opção abaixo:
encrypt passwords = false
passdb backend = smbpasswd guest

# Você pode usar um arquivo customizado de
# configuração deixando descomentada a linha abaixo, a macro %m deve ser
# substituída pelo nome da máquina que esta conectada.
; include = /home/samba/etc/smb.conf.%m

# Muitas pessoas afirmam que esta opção melhora a performance do
# Servidor, você pode querer adicionar a seguinte linha no seu Smb.Conf.
# SO_RCVBUF=8192 SO_SNDBUF=8192
socket options = TCP_NODELAY

#---------------------- Browser Control Options ----------------------

# Mude o Local Master para NO caso você não queira o samba como um
```

```
    # Master Browser na sua Rede
    local master = yes
    ; os level = 20

    # Dá suporte a logins de redes para o Windows 2000 e XP
    ; domain logons = yes

    # Não use esta opção caso você tenha um Domínio Windows NT fazendo
    # o trabalho de Domain Master
    ; domain master = auto
    ; preferred master = auto


    #--- End of Browser Control Options ---

    # seção de Suporte a Windows Internet Name Serving:
    # Wins Support - Diz ao componente do Samba NMBD para ativar este Wins Server
    ; wins support = yes# WINS Server - Diz ao NMBD para ser

    # um Cliente WINS Veja que: O Samba pode ser um WINS
    # Server, ou um WINS Client, mas não os 2
    ; wins server = w.x.y.z

    # Inclui informações do servidor DHCP, se este não estiver instalado
    # comente esta linha abaixo.
    include = /etc/samba/dhcp.conf

    # Isto previne o NMBD de procurar por nomes NetBios em DNS.
    dns proxy = no

    # Define o Chat para a alteração de senhas e define o programa de senhas
    passwd program = /usr/bin/passwd %u
    passwd chat = *InsirasUmasNovasSenha:* %n *RedigitesAsSenha:* %n .

    # Habilita logins com caracteres maiúsculos e minúsculos e mantém os # caracteres
dos nomes de arquivos na sua exata forma
    ; password level = 8
    ; preserve case = yes


    # Os seguintes parâmetros somente serão úteis caso você tenha o pacote
    # linpopup instalado
    ; message command = /bin/sh -c '/usr/bin/linpopup "%f" "%m" %s; rm %s' &
    obey pam restrictions = yes

    #==================== Share Definitions ====================

    [homes]
    comment = Diretórios Pessoais
    browseable = no

    # Por padrão os diretórios pessoais são exportados com o modo
    # somente-Leitura
    # Altere o próximo parâmetro para YES se você que deixá-los como Leitura e
    #Escrita
    writable = no

    # A criação de arquivos é setada como 0700 por razões de segurança se você
    # deseja criar arquivos com permissões group=rw, então marque o próximo
    # parâmetro como 0775
    create mask = 0700

    # A criação de diretórios usa como padrão permissões 0700 como medida de
    # segurança, caso você queira que as permissões sejam group=rw então
    # marque o parâmetro abaixo como 0775.
    directory mask = 0700

    # Descomente as linhas abaixo e crie o diretório NETLOGON para Domain
```

```
      #Logons
      # você terá que configurar o Samba para agir como um controlador de #domínios
também
      ;[netlogon]
      ; comment = Network Logon Service
      ; path = /home/samba/netlogon
      ; guest ok = yes
      ; writable = no
      ; share modes = no
      ; write list = ntadmin


      # Cria um compartilhamento de arquivos temporários, este está definido no
      #diretório /tmp, pode ser lido
      # e escrito, é publico.
      [tmp]
      comment = Arquivos temporários
      path = /tmp read only = no public = yes


      # Esta parte cria uma área pública a todos os usuários
      # Esta área fica no diretório /home/public (Você pode alterar o local se quiser)
      # Para os arquivos que estejam nesta área é permitido
      # a impressão a leitura e a impressão mesmo sendo um usuário com um ID
      #longe do root. [public]
      comment = Area Publica
      path = /home/public
      public = yes
      writable = yes
      printable = yes

      # A linha abaixo determina qual o grupo que tem permissão para escrever nos
      # Arquivos depositados neste diretório, altere o valor 'admin' para o grupo
      # Que você quer que tenha permissão para escrever
      # Nos arquivos.
      ; write list = @admin

      # Comente as linhas abaixo caso não queira compartilhar impressoras
      [printers]
      comment = Todas as Impressoras
      browseable = no
      path = /tmp
      printable = yes
      public = no
      writable = no
      create mode = 0700

      # Um exemplo para compartilhar seu CD-Rom com os outros usuários:
      # Pode ser que você precise criar o diretório /cdrom (Descomente as linhas
      #abaixo caso queira compartilhar)
      ;[cdrom]
      ; comment = CD-Rom em Servidor Samba
      ; writable = no
      ; locking = no
      ; path = /cdrom
      ; public = yes
      ; preexec = /bin/mount /cdrom
      ; postexec = /bin/umount /cdrom


      # Os próximos 2 parâmetros mostram como montar automaticamente um
      # CD-Rom quando este for acessado, para isto o arquivo /etc/fstab
      # precisa conter uma entrada como esta
      # /dev/scd0 /cdrom iso9660 defaults,noauto,ro,user 0 0
      # Isto se você possui um cdrom em /dev/scd0
      # Se você não quer usar o auto mount esteja certo que o CD esteja montado # em
/cdrom
```

2.7 Cups (Common Unix Printing System)

O Cups possui um recurso nativo de compartilhamento de impressoras. Ele permite não apenas compartilhar impressoras com outras máquinas Linux, mas também com máquinas Windows da rede, usando um servidor unificado. Para instalar o Cups use:

```
apt-get install cupsys
```

Compartilhar impressoras através do Cups é mais simples do que fazê-lo através do Samba e oferece uma vantagem adicional de permitir o uso do recurso de autodiscover do Cups nos clientes Linux.

O autodiscover permite que os clientes Linux da rede reconheçam automaticamente a impressora compartilhada e já a configurem durante o boot, sem necessidade de nenhuma intervenção manual.

Durante o boot o cliente manda um broadcast para a rede, solicitando para as máquinas presentes na rede, se alguma está compartilhando impressoras, caso exista um servidor de impressão instalado em alguma(s) maquina(s), o programa servidor responde que está compartilhando uma determinada impressora e já envia o driver usado pela impressora. Como ambos estão rodando o Cups significa que o cliente usa o mesmo conjunto de drivers de impressão do servidor, isso permite que ele simplesmente configure a impressora usando as informações recebidas, sem precisar perguntar nada ao usuário.

Caso você precise adicionar a impressora manualmente, execute o kaddprinterwizard (escolha no modo gráfico ou digite no terminal) e selecione a opção Remote CUPS Server. Forneça o endereço IP do servidor na rede locale a porta onde o Cups está escutando, que por padrão é a 631, ou então escolha uma impressora local.

Nos clientes Windows a configuração é semelhante. Eles não suportam o autodiscover por isso é preciso adicionar a impressora manualmente pelo Painel de Controle. Vamos supor que você deseja compartilhar a impressora hp. Na máquina servidora digite http://localhost:631, que automaticamente aparecerá a tela de configuração do Cups.

Acesse a opção Manage Printers e clique no link da impressora que será usada. Você verá um endereço como http://192.168.0.10:631/printers/hp (o IP informado é IP da máquina servidora de impressão) na barra do navegador. Este é o endereço da sua impressora, que vamos usar na instalação do Windows.

Dependendo da versão do Windows, quando for instalar a impressora, poderão ocorrer erros dizendo que não é possível se conectar à impressora. Dê um ok e volte à tela inicial. Marque agora a opção *Impressora local* e deixe marcado o *Detectar a instalar automaticamente impressora Plug and Play*. Ele dará outro erro, simplesmente confirme e diga que quer indicar a impressora manualmente. Você verá que apesar dos erros a impressora aparecerá disponível no final da lista. Basta selecioná-la e continuar com o processo normal de instalação da impressora, fornecendo o CD de drivers. O Cups é um servidor de impressão muito sólido, ele raramente dá problemas.

2.8 Servidor DHCP (Dynamic Host Configuration Protocol)

O DHCP é um serviço de rede que é responsável pela atribuição de endereços IP´s para todos os clientes da rede de forma automatizada. Isso possibilita a configuração rápida de todos clientes sem que seja necessária muita configuração local. Sempre que uma máquina cliente precisa de informações sobre a rede, a mesma fica enviando pacotes em broadcast até que o servidor DHCP retorna uma configuração para mesma.

Periodicamente o servidor DHCP verifica se as estações ainda estão lá, exigindo uma renovação do endereço IP (lease time). Assim os endereços IP são utilizados apenas com os equipamentos que estiver online, evitando que os endereços disponíveis se esgotem.

O deamon que executa o DHCP é o dhcp3-server pode ser instalado através do comando:

```
apt-get install dhcp3-server
```

O arquivo de script /etc/init.d/dhcp3-server é responsável pela operação do DHCP, já o arquivo de configuração é o /etc/dhcp3/dhcpd.conf. Um arquivo de configuração básico, contém o seguinte:

```
ddns-update-style none;
default-lease-time 600;
```

```
max-lease-time 7200;
authoritative;

subnet 192.168.0.0 netmask 255.255.255.0 {
       range 192.168.0.100 192.168.0.201;
       option routers 192.168.0.10;
       option domain-name-servers 200.177.250.10,200.204.0.10;
       option broadcast-address 192.168.0.255;
}
```

Para o exemplo acima, a opção `default-lease-time` controla o tempo de renovação dos endereços `IP`, nessa situação, o valor 600 indica que o servidor verifica a cada dez minutos (60 segundo X 10) se as estações ainda estão ativas. Se você tiver mais endereços `IP` do que máquinas os endereços IP das estações *raramente* vai precisar mudar. Mas, no caso de uma rede congestionada, a opção `max-lease-time` determina o tempo máximo que uma estação pode usar um determinado endereço `IP`.

A opção `range` determina a faixa de endereços `IP` que poderá ser usada pelo servidor. Já na `option routers` vai o endereço do `default gateway` da rede, ou seja, o endereço do servidor que está compartilhando a conexão.

Na opção `option domain-name-servers` contém os servidores `DNS` que serão usados pelas estações. Ao usar dois ou mais endereços eles devem ser separados por vírgula, sem espaços.

2.8.1 DHCP com IP fixo (análise de MAC)

Eventualmente, você pode desejar que algumas máquinas recebam sempre o mesmo `IP`, para liberar algumas portas e outros recursos. Para poder usar esta opção, acrescente uma seção como abaixo para cada máquina que desejar possuir `IP fixo`, para facilitar esse trabalho, anote o número `MAC` de cada estação. Adicione as linhas abaixo no final do arquivo `/etc/dhcpd.conf` (repita esse procedimento para cada máquina):

```
host atendimento {
       hardware ethernet 00:11:22:33:44:55;
       fixed-address 192.168.0.12;
}
```

Para descobrir o `MAC` das máquinas use o comando `ifconfig`. **Importante**: não use mais de um servidor `DHCP` na mesma faixa de rede, pois pode ocasionar conflitos de `IP`.

2.9 Servidor Squid

A palavra `Squid` (significa *lula,* um ser marinho com vários tentáculos que pega tudo que passa por perto).O `Squid` é um servidor `proxy-cache`, que permite compartilhar a conexão entre vários micros, servindo como um intermediário entre eles e a Internet. O `proxy` é um serviço que além de repassar requisições de conexões, ele analisa todo o tráfego de dados, separando o que pode ou não pode passar e guardando informações em `cache` para uso posterior.

Existem duas formas de se utilizar `proxy`, uma é usar o mesmo de forma concomitante com a conexão atual (onde você precisa especificar em cada máquina a configuração de `proxy`[6]), ou então usar `proxy` transparente. As principais vantagens de usar um `proxy` são:

- É possível impor restrições de acesso com base no horário, login, endereço IP da máquina e outras informações e bloquear páginas com conteúdo indesejado;
- O `proxy` funciona como um `cache` de páginas e arquivos, armazenando informações já acessadas. Quando alguém acessa uma página que já foi carregada, o `proxy` envia os dados que guardou no `cache`, sem precisar acessar a mesma página repetidamente. Isso acaba economizando banda na conexão com a Internet e tornando o acesso mais rápido sem precisar investir numa conexão mais rápida. Mesmo nas páginas dinâmicas, onde conteúdo muda a cada visita, o `proxy` agiliza pois geralmente são mantidas figuras, *banners*, animações em *flash* podem ser aproveitadas do `cache`, diminuindo o tempo total de carregamento. Dependendo da configuração, o proxy pode apenas

6 É necessário configurar o navegador e cada outro programa que for acessar a internet em cada cliente para usar o proxy. Esta é uma tarefa tediosa e que acaba dando bastante dor de cabeça a longo prazo, pois cada vez que um micro novo for colocado na rede será preciso fazer a configuração novamente. Além do navegador, outros programas, podem ser configurados para trabalhar através do proxy: Clientes de ICQ e MSN e até programas P2P.

acelerar o acesso às páginas, ou servir como um verdadeiro cache de arquivos, armazenando atualizações do Windows Update, downloads diversos e pacotes instalados através do apt-get por exemplo.

- Uma terceira vantagem de usar um proxy é que ele loga todos os acessos. Você pode visualizar os acessos posteriormente usando o Sarg.

Ao usar um `proxy` transparente, você tem basicamente uma conexão compartilhada via `NAT`, com a mesma configuração básica nos clientes. Uma regra de `iptables` envia as requisições recebidas na porta 80 do servidor para o `proxy`, que se encarrega de responder aos clientes. Toda a navegação passa a ser feita automaticamente através do `proxy`.

2.9.1 Instalando o Squid

A instalação do `Squid` é bastante simples e prática, pois o mesmo é composto de um único pacote, para instalar use:

```
apt-get install squid
```

Toda a configuração do `Squid` é feita num único arquivo, o `/etc/squid/squid.conf`. O arquivo original, instalado junto com o pacote é muito completo, contém comentários e exemplos para quase todas as opções disponíveis.

Apesar do `/etc/squid/squid.conf` ser muito extenso, é possível criar uma configuração relativamente simples. Observe o exemplo abaixo:

```
http_port 3128                    # executar na porta 3128
visible_hostname servidor         # nome do servidor
acl all src 0.0.0.0/0.0.0.0       # regra chamada all se refere a todos IP
http_access allow all             # diz que a regra all tem acesso permitido
```

Sempre que modificar qualquer linha no arquivo de configuração do Squid, você deverá reiniciar o serviço (*restart*):

```
/etc/init.d/squid
```

Caso desejar somente liberar algumas portas, você pode criar acl´s, cada acl é uma especificação de uma regra. O exemplo abaixo, mais completo, exemplifica o uso de várias acl´s:

```
http_port 3128                              # porta a ouvir
visible_hostname server                     # nome do servidor

acl all src 0.0.0.0/255.255.255.255        # regra all engloba todas redes
acl manager proto cache_object              # regra manager da placa local
acl localhost src 127.0.0.1/255.255.255.255 # regra localhost para IP local

acl SSL_ports port 443                      # regra SSL_ports composto de 443
acl SSL_ports port 563                      # inclui tambem a porta 563
acl Safe_ports port 80                      # regra Safe_ports para HTTP
acl Safe_ports port 21                      # inclui ftp
acl Safe_ports port 443 563                 # inclui https, snews
acl Safe_ports port 70                      # inclui gopher
acl Safe_ports port 210                     # inclui wais
acl Safe_ports port 1025-65535              # inclui portas acima de 1024
acl Safe_ports port 280                     # inclui http-mgmt
acl Safe_ports port 488                     # inclui gss-http
acl Safe_ports port 591                     # inclui filemaker
acl Safe_ports port 777                     # inclui multiling http
acl Safe_ports port 901                     # inclui SWAT

acl purge method PURGE                      # cria uma acl de eliminação
acl CONNECT method CONNECT                  # cria uma acl de conexão

http_access allow manager localhost         # permite acesso ao manager e localhost
http_access deny manager                    # proíbe os demais
```

```
http_access allow purge localhost          # permite eliminar sujeiras
http_access deny purge                     # proíbe os demais
http_access deny !Safe_ports               # bloqueia as portas que não são Safe
http_access deny CONNECT !SSL_ports        # bloqueia conectar se não for SSL

acl redelocal src 192.168.1.0/24           # cria regra para rede 1.0
http_access allow localhost                # permite acesso a localhost
http_access allow redelocal                # permite acesso a redelocal
http_access deny all                       # proíbe todas as redes
```

Depois de criadas as duas políticas de acesso, vão duas linhas no final do arquivo que especificam que os micros que se enquadrarem nelas vão poder usar:

```
http_access allow localhost
http_access allow redelocal
```

A cláusula `http_access deny all` diz que todos que pertencerem a *acl all* (todas interfaces de rede), estarão proibidas de acessar, no entanto como antes já informei quem pode, ela bloqueará todas redes menos as já liberadas.

O ordem das regras são de importância vital, pois o `Squid` interpreta as regras na ordem em que são colocadas no arquivo, caso você permite que o micro X acesse o `proxy`, ele acessará, mesmo que uma regra mais abaixo diga que não.

Para exemplificar a citação acima, vamos criar um exemplo onde primeiro uma rede será liberada, e logo em seguida bloqueada, nesse caso, como a permissão vem antes da proibição, a permissão é quem vigora. Observe o exemplo abaixo:

```
acl redelocal src 192.168.1.0/24           # regra redelocal para rede classe C
http_access allow redelocal                # permite acesso a redelocal
http_access deny redelocal                 # proíbe a acesso a redelocal
```

2.9.2 Melhorando as características do SQUID

O Squid trabalha com dois tipos de `cache`, um muito rápido que mantém as informações na memoria RAM e outro em disco rígido. A configuração da quantidade de memória RAM dedicada ao cache é feita adicionando a opção `cache_mem`, que contém a quantidade de memória que será dedicada ao `cache`. Para reservar 32 MB, por exemplo, a linha ficaria `cache_mem 32 MB`.

Para determinar o tamanho máximo dos arquivos que serão guardados no cache de memória RAM você deve utilizar `maximum_object_size_in_memory`. Para que o cache na memória armazene arquivos de até 64 KB por exemplo, adicione a linha `maximum_object_size_in_memory 64 KB`.

Em seguida vem a configuração do cache em disco, que armazenará a maior parte dos arquivos. Por default, o máximo são downloads de 16 MB e o mínimo é zero, o que faz com que mesmo imagens e arquivos pequenos sejam armazenados no cache.

Se você faz download de arquivos grandes e deseja que eles fiquem armazenados no cache, aumente o valor da opção `maximum_object_size` Isto é especialmente útil para quem precisa baixar muitos arquivos através do apt-get ou Windows update em muitos micros da rede. Se você quiser que o cache armazene arquivos de até 512 MB por exemplo, as linhas ficariam `maximum_object_size 512 MB` e para os menores `minimum_object_size 0 KB`.

É possível definir a percentagem de uso do cache que fará o squid começar a descartar os arquivos mais antigos. Por padrão isso começa a acontecer quando o cache está a 90% :

```
cache_swap_low 90               # começa a descartar
cache_swap_high 95              # descarta para atualizar
```

A configuração do tamanho do cache em disco propriamente dita é composta por quatro valores. O primeiro, `/var/spool/squid` indica a pasta onde o `Squid` armazena os arquivos do `cache`. Você pode querer alterar para uma pasta em uma partição separada por exemplo.

No exemplo abaixo, o valor 2048 indica a quantidade de espaço no HD (em MB) que será usada para o `cache`, os números 16 256 indicam a quantidade de subdiretórios que serão criadas dentro do diretório.

Por padrão temos 16 diretórios com 256 subdiretórios cada uma.

```
cache_dir ufs /var/spool/squid 2048 16 256
```

Você pode definir ainda o arquivo onde são guardados os `logs` de acesso do `Squid`. Por padrão o `log` é `/var/log/squid/access.log`. Este arquivo de `log`, é também usado pelo `sarg` para gerar as páginas com as estatísticas de acesso.

```
cache_access_log /var/log/squid/access.log
```

Para alterar o padrão de atualização do `cache`. Sempre modifique as três em conjunto.

```
refresh_pattern ^ftp: 15 20% 2280
refresh_pattern ^gopher: 15 0% 2280
refresh_pattern . 15 20% 2280
```

Os números indicam o tempo (em minutos) quando o `Squid` irá verificar se um ítem do `cache` foi atualizado, para cada um dos três protocolos. O primeiro número indica que o `Squid` verificará se todas as páginas e arquivos com mais de 15 minutos foram atualizados. Ele só verifica checando o tamanho do arquivo, o que é rápido. Se o arquivo não mudou, então ele continua mandando o que não está no `cache` para o cliente.

O número 2280 (equivalente a dois dias) indica o tempo máximo, depois disso o objeto é sempre verificado. Além do `http` e `ftp` o `Squid` suporta o protocolo `Gopher`, que era muito usado nos primórdios da Internet para localizar documentos de texto, mas perdeu a relevância hoje em dia.

Caso desejarmos criar uma arquivo de configuração usando o que foi mencionado acima:

```
http_port 3128
visible_hostname server
cache_mem 32 MB
maximum_object_size_in_memory 64 KB
maximum_object_size 512 MB
minimum_object_size 0 KB
cache_swap_low 90
cache_swap_high 95
cache_dir ufs /var/spool/squid 2048 16 256
cache_access_log /var/log/squid/access.log
refresh_pattern ^ftp: 15 20% 2280
refresh_pattern ^gopher: 15 0% 2280
refresh_pattern . 15 20% 2280
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 # https, snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl Safe_ports port 901 # SWAT
acl purge method PURGE
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
acl redelocal src 192.168.1.0/24
http_access allow localhost
http_access allow redelocal
```

```
http_access deny all
```

2.9.3 Bloqueando por palavras ou domínios

Uma forma fácil de bloquear sites no `Squid` é criar uma *lista de palavras*, um arquivo de texto onde você adiciona palavras e domínios que serão bloqueados no `Squid`.

Bloquear um determinado domínio, como por exemplo `orkut.com` não gera muitos problemas, mas tome cuidado ao bloquear palavras específicas, pois o `Squid` passará a bloquear qualquer página que contenha a palavra em questão.

Se você bloquear a palavra `sexo` por exemplo, qualquer site ou artigo que mencione a palavra será bloqueado. Ao bloquear por palavras você deve tentar ser específico, bloqueando apenas jargões e expressões que são encontradas apenas nos sites que você pretende bloquear. Para adicionar o filtro de palavras, adicione as linhas:

```
acl proibidos dstdom_regex "/etc/squid/proibidos"
http_access deny proibidos
```

Nesse caso, estamos criando uma acl chamada *proibidos* que é gerada a partir da leitura do arquivo /etc/squid/proibidos", o arquivo de texto que iremos editar. O acesso a qualquer página que contenha palavras citadas no arquivo é bloqueada. O arquivo pode conter palavras e domínios bloqueados, um por linha.

```
Orkut.com
playboy.com
lesbicas
```

Sempre que modificar alguma característica no arquivo ou arquivo de configuração do `Squid`, você deverá reiniciar o mesmo.

```
/etc/init.d/squid/restart
```

Caso queira, acrescente o código abaixo no seu arquivo funcional para barrar certos sites:

```
acl proibidos dstdom_regex "/etc/squid/proibidos"
http_access deny proibidos
acl bloqueados dstdomain orkut.com www.orkut.com playboy.abril.com.br
http_access deny bloqueados
acl redelocal src 192.168.1.0/24
http_access allow localhost
http_access allow redelocal
http_access deny all
```

2.9.4 Bloqueando por horário

É possível realizar bloqueios automatizados em determinados horários. Para que o `proxy` bloqueie acessos feitos entre meia-noite e 6:00 da manhã e no horário de almoço por exemplo:

```
acl madrugada time 00:00-06:00
http_access deny madrugada
acl almoco time 12:00-14:00
http_access deny almoco
```

Estas regras devem vir antes da regra `http_access allow redelocal` no arquivo de configuração. Outro exemplo:

```
acl almoco time 12:00-14:00
http_access allow almoco
```

Esta regra deve vir antes da regra `http_access deny proibidos` e `http_access deny proibidos`. Assim, os acessos que forem aceitos pela regra do almoço, não passarão pelas regras que fazem o bloqueio.

2.9.5 Proxy de Autenticação

Você pode adicionar uma camada extra de segurança exigindo autenticação no proxy. Este recurso pode ser usado para controlar quem tem acesso à Internet. Para ativar a autenticação você vai precisar de um programa chamado htpasswd. Se ele não estiver presente, instale o pacote apache-utils. Em seguida crie o arquivo que será usado para armazenar as senhas (caso não exista):

```
touch /etc/squid/squid_passwd
```

Após ter criado o arquivo acima, cadastre os logins usando o comando htpasswd /etc/squid/squid_passwd <Nome>, onde <Nome> é o nome do usuário que está sendo adicionado. Ex:

```
htpasswd /etc/squid/squid_passwd <Nome>
```

Depois de terminar de cadastrar os usuários, adicione as linhas que ativam a autenticação no /etc/squid/squid.conf:

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/squid_passwd
acl autenticados proxy_auth REQUIRED
http_access allow autenticados
```

Atenção, o /usr/lib/squid/ncsa_auth é a localização da biblioteca responsável pela autenticação. Eventualmente, ela pode estar numa pasta diferente, neste caso use o comando locate ou a busca do KDE para encontrar o arquivo e altere a linha indicando a localização correta.

2.9.6 Configurando um proxy transparente

Uma garantia de que os usuários realmente vão usar o proxy e ao mesmo tempo uma grande economia de trabalho e dor de cabeça pra você é o recurso de proxy transparente. Ele permite configurar o Squid de forma que o servidor proxy fique escutando todas as conexões na porta 80. Mesmo que alguém tente desabilitar o proxy manualmente nas configurações do navegador, ele continuará sendo usado.

Lembre-se que para usar o proxy transparente, você já deve estar compartilhando a conexão no servidor, via nat. O proxy transparente apenas fará com que o proxy intercepte os acessos na porta 80, obrigando tudo a passar pelas suas regras de controle de acesso, log, autenticação e cache.

Se você ainda não compartilhou a conexão, pode fazer isso manualmente rodando estes três comandos:

```
modprobe iptable_nat
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Imagine que a sua rede local esteja conectado ao servidor pela eth0, e indica a placa de rede que está conectada na Internet. Você pode checar a configuração da rede usando o comando ifconfig. Em seguida, rode o comando que direciona as requisições recebidas na porta 80 para o Squid.

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

Na linha acima, podemos ver que assim que chegam requisições (PREROUTING) na eth0 usando o protocolo tcp na porta 80, as mesmas são redirecionadas para o porta 3128 (porta de escuta do Squid). Para que essa sua configuração sempre seja executada ao reiniciar o servidor, adicione os quatro comandos no final do arquivo /etc/init.d/bootmisc.sh.

Finalmente, você precisa adicionar as seguintes linhas no final do arquivo /etc/squid/squid.conf e restartar o serviço:

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Os comandos acima, colocam o servidor como um gateway da rede para Squid com versões menores

a 2.6. Ao ativar o proxy transparente, a configuração dos clientes continuam igual, a única diferença é que agora todo o tráfego da porta 80 passará obrigatoriamente pelo servidor Squid.

Para versão 2.6 ou superior, o conjunto de comandos acima, devem ser descartados, no entanto, devemos modificar a primeira linha de configuração do SQUID, mais precisamente a linha que informa qual porta será utilizada, acrescentando a palavra *transparent*.

```
http_port 3128 transparent
```

Essa é a única modificação que devemos fazer no squid.conf para transformar o proxy em transparente.

2.10 Servidor Completo de E-mail's

Desenvolver um servidor de email completo, não é uma tarefa trivial, principalmente quando desejamos adicionar funcionalidades reais ao mesmo. Nessa seção será apresentado um esquema que permitirá realizar um servidor de alto padrão de funcionalidade, para isso, devemos ter suporte a domínios virtuais, cota de e-mail, anti-vírus, anti-spam, ferramentas de relatórios gráficas e em modo texto, autenticação SMTP (fundamental, por causa dos spammers), acesso POP3 e IMAP, tudo isso sendo gerenciado por uma ferramenta simples e fácil de se utilizar (ferramenta web usando php (seção 2.1.4).

Antes de começar a instalação propriamente dita, devemos conhecer alguns termos e protocolos fundamentais para melhor compreensão do que estamos produzindo.

- **MUA** (Mail User Agent): o MUA ou agente de usuário é um programa que permite aos usuários ler, salvar e compor mensagens de e-mail. Ex: ThunderBird, Kmail, Outlook Express entre outros.
- **MTA** (Mail Transfer Agent): o MTA é um software encarregado de direcionar, entregar e receber mensagens entre servidores. Um agente de transporte deve aceitar o e-mail de um agente de usuário (MUA), compreender os endereços dos receptores e de alguma maneira entregar o e-mail para os hosts corretos para entrega. Esses agentes de transporte, falam SMTP (Simple Mail Transfer Protocol) ou ESMTP (Extended Simple Mail Transfer Protocol), uma versão extendida do SMTP. Ex: Qmail, Postfix, Exim, SendMail entre outros.
- **MRA** (Mail Retrieval Agent (POP/IMAP client)): é um programa responsável por obter as mensagens de um servidor de e-mail. Ele realiza a conexão autenticada com o servidor POP ou IMAP e entrega as mensagens para serem filtradas pelo LDA. Ex: Fetchmail, Getmail, Retchmail entre outros.
- **LDA** (Local Delivery Agent): LDA ou agente de entrega aceita e-mail de um agente de transporte remetendo aos receptores locais apropriados. O LDA lê uma mensagem de e-mail da entrada padrão e entrega a mensagem para uma específica caixa de mensagem, de acordo com as instruções de seleção, filtragem e política anti-spam contidas em um arquivo de configuração. Um agente de entrega pode checar a sintaxe das regras contidas no arquivo de configuração e entregar a mensagem em caixas postais alternativas ou específicas. Ex: Procmail, Maildrop entre outros.

Pelo fato de não ser uma tarefa trivial, devemos instalar um conjunto de ferramentas para o nosso propósito. Assim sendo, execute o comando abaixo.

```
apt-get install mysql-server mysql-client libmysqlclient15-dev courier-imap
courier-authlib-mysql courier-imap-ssl courier-pop courier-pop-ssl gcc libc6-dev g++
libgdbm-dev gcc-3.4 cpp make postfix postfix-mysql sasl2-bin libsasl2-modules-sql libpam-
mysql clamav-daemon clamav-freshclam amavisd-new phpmyadmin php4 apache php4-mysql
libapache-mod-php4 libgsasl7 libsasl2 libsasl2-dev libsasl2-modules libfile-mmagic-perl
libconfig-inifiles-perl libconvert-tnef-perl libconvert-uulib-perl libio-zlib-perl
libarchive-tar-perl libarchive-zip-perl libparse-syslog-perl libunix-syslog-perl libmime-
perl libmime-base32-perl libfile-mimeinfo-perl libnet-server-perl libnet-smtp-server-perl
libmd5-perl ncftp unzip ftp gnupg arj cabextract
```

Analisando os pacotes acima, podemos observar a existência do servidor de banco de dados mysql (seção 2.2) (será utilizado para gerenciar os usuários, bem como, email's enviados e recebidos, em outras palavras, a estrutura como um todo será armazenada), o servidor courier, tanto para IMAP quanto POP, ferramentas para eventuais compilações, o servidor postfix (SMTP), clamav (anti-virus), amavis (anti-spam), phpmyadmin (gerenciador mysql via web), servidor apache (seção 2.1), perl, utilitários de compactação e vários módulos e bibliotecas necessárias.

Como o gerenciamento e configuração de um servidor de email's é um processo relativamente

complexo, foram criadas ferramentas para auxiliar esses procedimentos. Para instalar as mesmas, você deve procurar na internet por *postfixadmin* e o *maildrop.*

Caso deseje, pode tentar usar um site da sourceforge, utilizando os comandos abaixo (pode acontecer que a versão abaixo não exista mais). Idealmente, execute os comandos abaixo estando no diretório /usr/local/src ou então faça o download e posteriormente os copie para esse diretório.

```
wget -c http://high5.net/postfixadmin/download.php?file=postfixadmin-2.1.0.tgz -O postfixadmin.tgz
wget -c http://umn.dl.sourceforge.net/sourceforge/courier/maildrop-1.6.3.tar.bz2 -O maildrop.tar.bz2
```

2.10.1 Ajustes no mysql

Supondo que você esteja instalando pela primeira vez o mysql, realize o passo abaixo, pois o mesmo definirá a senha do mysql (senha de administrador).

```
mysqladmin -u root password 'suasenha'
```

Outro fator importante é habilitar os log's gerados pelo mesmo, eles servirão para posteriormente analisarmos a utilização do servidor de email pelos usuários. Para isso edite o arquivo */etc/mysql/my.cnf* e descomente a seguinte linha (retire #):

```
#log            = /var/log/mysql.log
```

Sempre que modificamos um arquivo de configuração de um programa servidor, devemos restartá-lo.

```
/etc/init.d/mysql restart
```

Nesse ponto falta ainda criar um usuário para gerenciar o *maildrop*, para isso execute o comando abaixo (anote a senha, pois será solicitada abaixo):

```
adduser maildrop
```

Importante, ao executar o comando acima, será apresentado na tela algo similar:

```
Acrescentando usuário maildrop...
Adding new group `maildrop' (1003).
Adding new user `maildrop' (1003) with group `maildrop'.
Criando diretório pessoal /home/maildrop.
Copiando arquivos de /etc/skel
Enter new UNIX password:
```

Anote o número que aparecer no grupo e no usuário (geralmente iguais(padrão Debian)), pois esses valores serão utilizados pelo procedimento a seguir.

2.10.2 Criando a estrutura de gerenciamento do postfix no MySQL

Nesse ponto começa a árdua tarefa de criar a estrutura de banco de dados necessária para armazenar os email's e suas necessidades especiais. Substitua no arquivo abaixo as linhas:

```
uid int(1003) unsigned default '1003',
gid int(1003) unsigned default '1003',
```

onde, devemos trocar 1003 pela ID do usuário (UID) e grupo (GID) maildrop criados no procedimento anteriormente.

Caso deseje, pode colocar todos os comandos abaixo em um arquivo texto, e depois executar, para isso use:

```
mysql -u root -p < ARQUIVOCOMOSCOMANDOS
```

A senha solicitada será a que você cadastrou como senha do mysql. Em seguida será apresentado todos os comandos que o usuário deverá executar para criar a estrutura do mysql. Sempre que uma linha começar com #, significa que essa linha é um comentário, assim sendo, não é necessário digitar a mesma (Substitua

no script abaixo xxxxx por sua senha.

```
#Inicio do script utilizado para criar a estrutura necessária no mysql
USE mysql
#Cria o usuÃ¡rio e senha do Postfix e Maildrop para poder acessar o banco de dados
#substitua o  xxxxx abaixo pela senha que você deseja para o postfix
INSERT INTO user (Host, User, Password) VALUES('localhost','postfix',password('xxxxx'));
#substitua o  xxxxx abaixo pela senha que você deseja para o maildrop (pode ser a mesma que já foi utilizada anteriormente)
INSERT INTO user (Host, User, Password) VALUES('localhost','maildrop',password('xxxxx'));
INSERT INTO db (Host, Db, User, Select_priv) VALUES ('localhost','postfix','postfix','Y');
INSERT INTO db (Host, Db, User, Select_priv) VALUES ('localhost','postfix','maildrop','Y');
FLUSH PRIVILEGES;

# Cria o banco postfix
CREATE DATABASE postfix;
# Cria a estrutura da tabela alias
USE postfix;
CREATE TABLE alias (address varchar(255) NOT NULL default '', goto text NOT NULL, domain varchar(255) NOT NULL default '', create_date datetime NOT NULL default '0000-00-00 00:00:00', change_date datetime NOT NULL default '0000-00-00 00:00:00', active tinyint(4) NOT NULL default '1', PRIMARY KEY (address)) TYPE=MyISAM COMMENT='Virtual Aliases - mysql_virtual_alias_maps';
# Cria a estrutura da tabela domain
CREATE TABLE domain (domain varchar(255) NOT NULL default '', description varchar(255) NOT NULL default '', transport varchar(255) NOT NULL default 'maildrop', create_date datetime NOT NULL default '0000-00-00 00:00:00', change_date datetime NOT NULL default '0000-00-00 00:00:00', active tinyint(4) NOT NULL default '1', PRIMARY KEY (domain)) TYPE=MyISAM COMMENT='Virtual Domains - mysql_virtual_domains_maps';
CREATE TABLE mailbox (username varchar(255) NOT NULL default '',password varchar(255) NOT NULL default '', name varchar(255) NOT NULL default '', home char(255) default '/postfix/', maildir varchar(255) NOT NULL default '', quota varchar(255) NOT NULL default '10000000S', domain varchar(255) NOT NULL default '', create_date datetime NOT NULL default '0000-00-00 00:00:00', change_date datetime NOT NULL default '0000-00-00 00:00:00', active tinyint(4) NOT NULL default '1', passwd_expire enum('N','Y') default 'Y', uid int(10) unsigned default '1001', gid int(10) unsigned default '1001',PRIMARY KEY (username)) TYPE=MyISAM COMMENT='Virtual Mailboxes - mysql_virtual_mailbox_maps';

# Cria o usuário de administração do PostfixAdmin
USE mysql
INSERT INTO user (Host, User, Password) VALUES ('localhost','postfixadmin',password('xxxxx'));
INSERT INTO db (Host, Db, User, Select_priv, Insert_priv, Update_priv, Delete_priv)VALUES ('localhost', 'postfix', 'postfixadmin', 'Y', 'Y', 'Y', 'Y');
FLUSH PRIVILEGES;

# Cria a tabela de administração do PostfixAdmin
USE postfix;
CREATE TABLE admin (username varchar(255) NOT NULL default '',password varchar(255) NOT NULL default '', domain varchar(255) NOT NULL default '', create_date datetime NOT NULL default '0000-00-00 00:00:00', change_date datetime NOT NULL default '0000-00-00 00:00:00', active tinyint(4) NOT NULL default '1', PRIMARY KEY (username)) TYPE=MyISAM COMMENT='Virtual Admins - Store Virtual Domain Admins';
```

Executados todos os comandos, e sem erro, à principio estará tudo correto, no entanto, vamos testar a funcionalidade executando o seguinte comando:

```
mysql -D postfix -u postfix -p senha_do_postfix
```

Se aparecer uma informação similar a abaixo, o serviço estará funcionando:

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Welcome to the MySQL monitor.  Commands end with ; or g.
Your MySQL connection id is 7 to server version: 5.0.22-Debian_3-log
Type 'help;' or 'h' for help. Type 'c' to clear the buffer.
```

Execute o comando abaixo para ter certeza da estrutura criada:

```
mysql> show tables;
```

Deverá aparecer a seguinte estrutura:

```
+--------------------+
| Tables_in_postfix |
+--------------------+
| admin              |
| alias              |
| domain             |
| mailbox            |
+--------------------+
4 rows in set (0.00 sec)
```

Execute o comando abaixo para ver a característica da estrutura da tabela alias:

```
mysql> desc alias;
```

O resultado deverá ser:

```
+-------------+--------------+------+-----+---------------------+-------+
| Field       | Type         | Null | Key | Default             | Extra |
+-------------+--------------+------+-----+---------------------+-------+
| address     | varchar(255) | NO   | PRI | NULL                |       |
| goto        | text         | NO   |     | NULL                |       |
| domain      | varchar(255) | NO   |     | NULL                |       |
| create_date | datetime     | NO   |     | 0000-00-00 00:00:00 |       |
| change_date | datetime     | NO   |     | 0000-00-00 00:00:00 |       |
| active      | tinyint(4)   | NO   |     | 1                   |       |
+-------------+--------------+------+-----+---------------------+-------+
6 rows in set (0.00 sec)
```

A tabela alias é o local onde serão criados os redirecionamentos de e-mail. Exemplo:

```
postmaster@dominio.com.br > usuario@dominio.com.br
postmaster@dominio2.com.br => usuario@dominio2.com.br, usuario1@dominio1.com.br
```

Execute o comando abaixo para ver a característica da estrutura da tabela de domínios:

```
mysql> desc domain;
```

O resultado deverá ser:

```
+-------------+--------------+------+-----+---------------------+-------+
| Field       | Type         | Null | Key | Default             | Extra |
+-------------+--------------+------+-----+---------------------+-------+
| domain      | varchar(255) |      | PRI |                     |       |
| description | varchar(255) |      |     |                     |       |
| transport   | varchar(128) |      |     | maildrop            |       |
| create_date | datetime     |      |     | 0000-00-00 00:00:00 |       |
| change_date | datetime     |      |     | 0000-00-00 00:00:00 |       |
| active      | tinyint(4)   |      |     | 1                   |       |
+-------------+--------------+------+-----+---------------------+-------+
```

Execute o comando abaixo para ver a característica da estrutura da tabela mailbox:

```
mysql> desc mailbox;
```

O resultado deverá ser:

```
+--------------+-----------------+------+-----+---------------------+-------+
```

```
| Field         | Type             | Null | Key | Default             | Extra |
+---------------+------------------+------+-----+---------------------+-------+
| username      | varchar(255)     |      | PRI |                     |       |
| password      | varchar(255)     |      |     |                     |       |
| name          | varchar(255)     |      |     |                     |       |
| home          | varchar(255)     | YES  |     | /postfix/           |       |
| maildir       | varchar(255)     |      |     |                     |       |
| quota         | varchar(255)     |      |     |10000000S            |       |
| domain        | varchar(255)     |      |     |                     |       |
| create_date   | datetime         |      |     | 0000-00-00 00:00:00 |       |
| change_date   | datetime         |      |     | 0000-00-00 00:00:00 |       |
| active        | tinyint(4)       |      |     | 1                   |       |
| uid           | int(10) unsigned | YES  |     | 108                 |       |
| gid           | int(10) unsigned | YES  |     | 108                 |       |
| passwd_expire | enum('N','Y')    | YES  |     | Y                   |       |
+---------------+------------------+------+-----+---------------------+-------+
```

Essa é a principal tabela, pois nela serão cadastrados os usuários e suas configurações, como por exemplo, senha, diretório, cota de disco, entre outras características.

Execute o comando abaixo para ver a característica da estrutura da tabela admin:

```
mysql> desc admin;
```

O resultado deverá ser:

```
+-------------+--------------+------+-----+---------------------+-------+
| Field       | Type         | Null | Key | Default             | Extra |
+-------------+--------------+------+-----+---------------------+-------+
| username    | varchar(255) |      | PRI |                     |       |
| password    | varchar(255) |      |     |                     |       |
| domain      | varchar(255) |      |     |                     |       |
| create_date | datetime     |      |     | 0000-00-00 00:00:00 |       |
| change_date | datetime     |      |     | 0000-00-00 00:00:00 |       |
| active      | tinyint(4)   |      |     | 1                   |       |
+-------------+--------------+------+-----+---------------------+-------+
```

2.10.3 Ajustando o Courier

Para facilitar a configuração do courier é altamente recomendável pegar uma estrutura padrão, para isso iremos copiar um o arquivo de exemplo de *warning* do cota:

```
cd /etc/courier
cp /usr/share/doc/courier-base/examples/quotawarnmsg.example quotawarnmsg
```

Após isso, devemos nos deter na autenticação. As estrutura de configuração default dos arquivos *imapd* e *pop3d* já são funcionais, no entanto, a medida que for tendo mais intimidade com o servidor de email, é altamente recomendável ajustar alguns itens para obter melhor desempenho.

Edite o */etc/courier/authmysqlrc*. Esse arquivo é responsável pela configuração do Courier, em outras palavras, é o responsável para fazer a conexão ao mysql, de forma a autenticar os usuários. Ajuste esse arquivo de acordo com sua necessidade (mudar xxxx).

```
MYSQL_SERVER            localhost
MYSQL_USERNAME          postfix
MYSQL_PASSWORD          xxxxx
MYSQL_SOCKET            /var/run/mysqld/mysqld.sock
MYSQL_PORT              3306
MYSQL_OPT               0
MYSQL_DATABASE          postfix
MYSQL_USER_TABLE        mailbox
MYSQL_CRYPT_PWFIELD     password
MYSQL_UID_FIELD         uid
MYSQL_GID_FIELD         gid
MYSQL_LOGIN_FIELD       username
MYSQL_HOME_FIELD        home
```

```
MYSQL_NAME_FIELD        name
MYSQL_MAILDIR_FIELD     maildir
MYSQL_QUOTA_FIELD       quota
MYSQL_WHERE_CLAUSE      active=1
```

Após tudo realizado, é altamente importante reiniciar os serviços, para isso use (padrão Debian):

```
/etc/init.d/courier-authdaemon restart
/etc/init.d/courier-imap restart
/etc/init.d/courier-pop restart
```

Para testar o funcionamento dos servidores execute um *telnet* nas suas portas:

```
telnet 143
telnet 110
```

2.10.4 Configuração do maildrop

Você deverá criar o arquivo */etc/maildropmysql.config*, os comandos de configuração podem ser os seguintes, os mesmos serão utilizados para compilar e instalar o *maildrop.*

```
hostname                localhost
port                    3306
socket                  /var/lib/mysql/mysql.sock
database                postfix
dbuser                  maildrop
dbpw                    suasenha
dbtable                 mailbox
default_gidnumber       seuGID
default_uidnumber       seuUID
uid_field               username
uidnumber_field         uid
gidnumber_field         gid
maildir_field           maildir
homedirectory_field     home
quota_field             quota
mailstatus_field        active
where_clause            ""
```

Antes de realizar a configuração devemos instalar o mesmo, na seção 2.10, foi realizado o download do maildrop, o qual foi copiado para o diretório */usr/local/src*. Nesse momento iremos descompactar, compilar o mesmo.

```
tar xvjf maildrop.tar.bz2
cd maildrop
./configure
        --prefix=/usr
        --sysconfdir=/etc/maildrop
        --enable-maildrop-uid=seuUID
        --enable-maildrop-gid=seuGID
        --enable-syslog=1
        --enable-maildropmysql
        --enable-maildirquota
```

Vamos agora ao processo de compilação, para isso digite:

```
make && make install
```

O *Maildrop* possui um recurso de aviso de email para o usuário caso a sua caixa postal ultrapasse a porcentagem de utilização de um valor especificado. Nesse caso será utilizado o valor de 90%. Assim sendo, você poderá personalizar a mensagem de aviso que o usuário irá recebe, bastando para isso editar o arquivo:

```
/etc/courier/quotawarnmsg.
```

2.10.5 Configuração do Postfix

Primeiramente devemos verificar se o pacote instalado no sistema realmente tem suporte ao MySQL, ou seja possui o módulo de acesso, para isso digite no diretório */etc/courier*:

```
postconf -m
 btree
 cidr
 environ
 hash
 mysql
 nis
 proxy
 regexp
 sdbm
 static
 tcp
 unix
```

Caso possua o mysql, podemos passar para o próximo passo. Entre no diretório */etc/postfix/*.

Primeiramente criaremos o arquivo *mysql_virtual_alias_maps.cf* com o seguinte conteúdo (substitua xxxx pela senha que cadastrou para o usuário postfix):

```
user = postfix
password = xxxxx
dbname = postfix
table = alias
select_field = goto
where_field = address
hosts = localhost
```

Agora criaremos o arquivo *mysql_virtual_mailbox_maps.cf* com o seguinte conteúdo (substitua xxxx pela senha que cadastrou para o usuário postfix):

```
user = postfix
password = xxxxx
dbname = postfix
table = mailbox
select_field = maildir
where_field = username
hosts = localhost
```

Agora criaremos o arquivo *mysql_transport_maps.cf* com o seguinte conteúdo (substitua xxxx pela senha que cadastrou para o usuário postfix):

```
user = postfix
password = xxxxx
hosts = localhost
dbname = postfix
table = domain
select_field = transport
where_field = domain
```

Agora criaremos o arquivo *mysql_virtual_mailbox_limit_maps.cf* com o seguinte conteúdo (substitua xxxx pela senha que cadastrou para o usuário postfix):

```
user = postfix
password = xxxxx
dbname = postfix
table = mailbox
select_field = quota
where_field = username
hosts = localhost
```

Nesse momento precisamos editar o arquivo *master.cf,* edite todos os parâmetros da coluna *chroot* para *n*, pois não será utilizado o *postfix* em um ambiente *chroot.* Ajuste o suporte ao Maildrop:

```
maildrop  unix  -       n       n       -       -       pipe
   flags=DRhu user=maildrop argv=/usr/bin/maildrop -w 90 -d ${recipient}
```

A configuração acima deve ser identada, assim deixe 3 espaços no inicio da linha. O parâmetro "-w 90" informa a porcentagem de utilização que a caixa postal pode chegar. A mensagem de aviso é enviada pelo *MAILDROP* com o conteúdo do arquivo */etc/courier/quotawarnmsg*.

Agora editaremos o arquivo *main.cf*, mas antes faremos uma cópia do arquivo já existente, pois o mesmo possui as opções possíveis, como são muitas, criaremos um arquivo do zero.

```
cp main.cf mainf.cf.bkp
> main.cf
```

Edite o arquivo zerado e acrescente os seguintes comandos, substituindo os valores em negrito de acordo com a sua necessidade:

```
queue_directory = /var/spool/postfix/
program_directory=/usr/sbin
command_directory = /usr/sbin
daemon_directory = /usr/lib/postfix
mail_owner = postfix
default_privs=nobody
default_transport=smtp
local_recipient_maps =
delay_warning_time = 5m
alias_maps=hash:/etc/postfix/aliases
alias_database=hash:/etc/postfix/aliases
readme_directory = no
sample_directory = /etc/postfix
sendmail_path = /usr/sbin/sendmail
setgid_group = postdrop
manpage_directory = /usr/local/man
newaliases_path = /usr/bin/newaliases
mailq_path = /usr/bin/mailq
smtpd_banner=$myhostname ESMTP MEU_DOMINIO
disable_vrfy_command=yes
home_mailbox=Maildir/

myhostname=postfix.dominio.com.br
mydomain=local.com.br
myorigin= $mydomain
mydestination= $mydomain, $transport_maps

mynetworks=127.0.0.0/8 192.168.0.0/24

virtual_alias_maps = mysql:/etc/postfix/mysql_virtual_alias_maps.cf
virtual_mailbox_base = /postfix
virtual_mailbox_maps = mysql:/etc/postfix/mysql_virtual_mailbox_maps.cf
virtual_uid_maps = static:seuUID
virtual_gid_maps = static:seuGID
transport_maps = mysql:/etc/postfix/mysql_transport_maps.cf

virtual_mailbox_limit_inbox = no
virtual_mailbox_limit_maps= mysql:/etc/postfix/mysql_virtual_mailbox_limit_maps.cf
virtual_mailbox_limit_override = yes
virtual_maildir_extended = yes
virtual_create_maildirsize = yes
virtual_mailbox_limit = 100000000
virtual_maildir_limit_message = Sua cota de disco se foi, tente mais tarde.
virtual_overquota_bounce = yes

#smtpd_sasl_auth_enable = yes
#smtpd_sasl_security_options = noanonymous
#broken_sasl_auth_clients = yes
#smtpd_recipient_restrictions =
```

```
#permit_sasl_authenticated,
#permit_mynetworks,
#check_relay_domains

fallback_transport = /usr/bin/maildrop
maildrop_destination_recipient_limit = 1
unknown_local_recipient_reject_code = 450
```

Note que algumas linhas estão comentadas, elas serão úteis durante a sua configuração. O parâmetro *maildrop_destination_recipient_limit* faz com que o *MAILDROP* entregue os email's para mais de um destinatária, caso contrario, apenas um vai receber a mensagem.

Precisamos agora ajustar os direitos do usuário em relação ao mysql, para isso devemos nos logar como root no MySQL:

```
mysql -u root -p mysql
```

E executar os seguintes comandos:

```
UPDATE `db` SET `Insert_priv` = 'Y',`Update_priv` = 'Y',`Delete_priv` = 'Y'
WHERE CONVERT( `Host` USING utf8 ) = 'localhost' AND CONVERT( `Db` USING utf8 ) =
'postfix' AND CONVERT( `User` USING utf8 ) = 'postfix' LIMIT 1;
flush privileges;
```

Vamos agora logar no banco MySQL e inserir um usuário e um domínio para testes;

```
mysql -D postfix -u postfix -p suasenha
```

2.10.5.1 Teste manual dos serviços

Nesse ponto podemos realizar um teste *manual* (para não dizer braçal) para ver o funcionamento dos serviços envolvidos. Inserindo domínio para testes:

```
INSERT INTO domain (domain, description, transport, active) VALUES
('dominio1.com.br' , 'Dominio de Teste' , 'maildrop' ,'1');
```

Verificando domínio inserido:

```
SELECT domain, description, transport, active FROM domain;
```

Deverá aparecer:

```
+-----------------+--------------------------+-----------+--------+
| domain          | description              | transport | active |
+-----------------+--------------------------+-----------+--------+
| dominio1.com.br | Dominio de Teste         | maildrop  |      1 |
+-----------------+--------------------------+-----------+--------+
```

Inserindo um usuário para testes:

```
INSERT INTO mailbox (username, password, name, home, maildir, quota, domain)VALUES
('usuario.silva@dominio1.com.br' , encrypt('xxxxx') ,'Usuario da Silva' , '/postfix/' ,
'dominio1.com.br/usuario.silva/Maildir/' , '10000000S' ,'dominio1.com.br');
```

Verificando o usuário criado:

```
SELECT username FROM mailbox;
```

Deverá aparecer:

```
+------------------------------+
| username                     |
+------------------------------+
| usuario.silva@dominio1.com.br |
+------------------------------+
```

Criando o *HOME* do usuário:

```
mkdir -p /postfix/dominio1.com.br/usuario.silva
maildirmake /postfix/dominio1.com.br/usuario.silva/Maildir
chown maildrop:www-data /postfix/ -R
chmod 770 /postfix/ -R
```

Testando a entrega de mensagens para o usuário. Para realizar essa tarefa, primeiramento devemos verificar se o *maildrop* consegue se comunicar com o MySQL, usando o comando:

```
maildrop -v
```

O resultado desejado deverá ser similar a:

```
GDBM extensions enabled.
Maildir quota extension enabled.
Virtual user database via MySQL extension enabled.
This program is distributed under the terms of the GNU General Public License. See
COPYING for additional information.
```

Execute o comando abaixo para poder enviar uma mensagem de modo direto:

```
cat /etc/lilo.conf| maildrop -d usuario.silva@dominio1.com.br
```

Conferindo se a mensagem foi entregue:

```
cd /postfix/dominio1.com.br/usuario.silva/Maildir/new
ls
```

Deverá aparecer algo como:

```
1154248310.M755807P1285V0000000000000802I00006692_0.smtp.catoca.com,S=4127
```

2.10.5.2 Testando o IMAP via banco de dados:

Para testar se o protocolo está corretamente configurado, podemos executar um `telnet` na sua porta padrão, para isso:

```
telnet localhost 143
```

O resultado deverá ser algo como:

```
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT
THREAD=REFERENCES SORT QUOTA IDLE ACL ACL2=UNION STARTTLS] Courier-IMAP ready.
Copyright 1998-2005 Double Precision, Inc. See COPYING for distribution
information.
```

digite:

```
login usuario.silva@dominio1.com.br minhasenha
```

O resultado deverá ser algo como:

```
0 OK LOGIN Ok.
0 select inbox
* FLAGS (Draft Answered Flagged Deleted Seen Recent)
* OK [PERMANENTFLAGS (* Draft Answered Flagged Deleted Seen)] Limited
* 1 EXISTS
* 1 RECENT
* OK [UIDVALIDITY 1154248579] Ok
* OK [MYRIGHTS "acdilrsw"] ACL
0 OK [READ-WRITE] Ok
```

2.10.5.3 Testando a configuração do POP:

Para testar se o protocolo está corretamente configurado, podemos executar um `telnet` na sua porta padrão, para isso:

```
# telnet localhost 110
```

Digite:

```
user usuario.silva@dominio1.com.br
pass minhasenha

list
```

Como resultado deverá aparecer algo como:

```
+OK POP3 clients that break here, they violate STD53.
```

2.10.6 Cota de e-mails

O `maildrop` vai ser responsável pelo controle de cota de disco. Os valores de cotas serão cadastrados no banco de dados, sendo independente para cada usuário. O `maildrop` lerá as tabelas do `MySQL` e de acordo com os valores da configuração, vai ajustar as cotas de usuário.

Toda vez que um usuário recebe um e-mail, o Maildrop calcula o espaço utilizado. Para consultar o valor da cota do usuário, execute o seguinte comando no shell (entrando no mysql).

```
mysql -D postfix -u postfix -psuasenha -e "SELECT username,quota FROM mailbox
WHERE username='usuario.silva@dominio1.com.br';"
```

Como resultado teremos (dado que tenhamos cadastrado o usuário acima):

```
+------------------------------+-----------+
| username                     | quota     |
+------------------------------+-----------+
| usuario.silva@dominio1.com.br | 10000000S |
+------------------------------+-----------+
```

Nesse exemplo a cota esta em 10MB, vamos altera-la para 10KB, com o seguinte comando:

```
mysql -D postfix -u postfix -psuasenha -e "UPDATE mailbox SET quota='10000S'
WHERE username='usuario.silva@dominio1.com.br';"
```

2.10.7 SASL2 no Postfix:

O `SASL` permite que um usuário consiga enviar e-mail pelo servidor `SMTP` (relay) sem que o seu IP esteja na lista de IP's liberados para relay, no postfix é configurado na linha "*mynetworks"* no arquivo *main.cf*. O requisito para enviar e-mail é que o usuário exista no sistema. Isso é um ótimo recurso, pois o usuário onde estiver pode enviar e-mail pelo seu servidor, sem que você precise liberar o "*Relay"* para todo mundo.

Crie arquivo `/usr/lib/sasl2/smtpd.conf` com o seguinte conteúdo:

```
pwcheck_method: saslauthd
```

Crie arquivo `/etc/pam.d/smtp` com o seguinte conteúdo:

```
auth     sufficient   /lib/security/pam_unix_auth.so try_first_pass
auth     optional   /lib/security/pam_mysql.so user=postfix
                    passwd=xxxxx
                    host=localhost
                    db=postfix
                    table=mailbox
                    usercolumn=username
                    passwdcolumn=password
```

```
                    crypt=1
account  sufficient   /lib/security/pam_unix_acct.so
account     required   /lib/security/pam_mysql.so user=postfix
                    passwd=xxxxxx
                    host=localhost
                    db=postfix
                    table=mailbox
                    usercolumn=username
                    passwdcolumn=password
                    crypt=1
```

Para habilitar o SASL no Postfix, descomente as seguintes linhas no arquivo /etc/postfix/main.cf:

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
   permit_sasl_authenticated,
   permit_mynetworks,
   check_relay_domains
```

Edite o arquivo /etc/default/saslauthd, e descomente a linha:

```
# START=yes
```

Reinicie os serviços:

```
/etc/init.d/saslauthd start
/etc/init.d/postfix restart
```

2.10.8 Configurando o Amavis

A linguagem Perl já deve estar instalada, assim nós devemos ajustar as dependências do Perl com o Perl-CPAN. Para isso execute (responda todas as respostas padrão, e as que se refere ao país e outras mais simples, escolha de acordo com as necessidades):

```
perl -MCPAN -e shell
```

Agora dentro do Perl, execute:

```
cpan> install File::MMagic
cpan> install Config::IniFiles
cpan> install Convert::TNEF
cpan> install Convert::UUlib
cpan> install Compress::Zlib
cpan> install Archive::Tar
cpan> install Archive::Zip
cpan> install Unix::Syslog
cpan> install MIME::Base64
cpan> install Net::Server
cpan> install Net::SMTP

           Should all FTP connections be passive (y|n) ? [no] no

cpan> install Digest::MD5
cpan> install Time::HiRes
cpan> install Mail::SpamAssassin
cpan> install Digest::SHA1
cpan> install HTML::Parser
cpan> install Net::DNS
cpan> install IP::Country
cpan> install Net::Ident
cpan> install LWP::UserAgent
cpan> install HTTP::Date
cpan> exit
```

Adicione no /etc/postfix/master.cf o seguinte bloco de texto:

```
smtp-amavis unix -      -       n       -       2  smtp
        -o smtp_data_done_timeout=1200
        -o disable_dns_lookups=yes
127.0.0.1:10025 inet n -       n       -       -  smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_client_restrictions=
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o mynetworks=127.0.0.0/8
```

Edite o arquivo /etc/amavis/conf.d/15-content_filter_mode, e descomente os seguintes blocos de texto:

```
#@bypass_virus_checks_maps = (
#   %bypass_virus_checks, @bypass_virus_checks_acl, $bypass_virus_checks_re);
#@bypass_spam_checks_maps = (
#   %bypass_spam_checks, @bypass_spam_checks_acl, $bypass_spam_checks_re);
```

Isso habilita a checagem contra vírus e spams. Adicione o amavis no grupo do clamav, e vice-versa, isso é necessário para que eles possam acessar um os arquivos do outro, para isso execute:

```
adduser clamav amavis
adduser amavis clamav
```

Inicie o serviço do amavis e do clamav:

```
/etc/init.d/amavis start
/etc/init.d/clamav-daemon restart

Adicione ao final do /etc/postfix/main.cf:

content_filter = smtp-amavis:[127.0.0.1]:10024
```

Reinicie o serviço do postfix:

```
/etc/init.d/postfix restart
```

Vamos agora fazer um teste, para saber se nosso servidor realmente está evitando vírus, vamos utilizar para isto o arquivo de teste do EICAR, com o seguinte comando:

```
mail -s "teste" usuario.silva@dominio.com.br

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

2.10.9 Configurando o PostfixAdmin

Primeiramente devemos instalar alguns pacotes auxiliares, dessa forma, digite:

```
apt-get update
apt-get install pflogsumm procmail
```

Edite o arquivo /etc/logrotate.conf, e adicione o seguinte bloco de texto:

```
/var/log/mail.log {
    missingok
    daily
    rotate 7
    create
```

```
        compress
        start 0
    }
```

Isso garante o rotacionamento do log uma vez por dia, compactando os arquivos antigos do log. Crie o arquivo /usr/local/sbin/postfix_report.sh, que será utilizado para a geração de relatórios e envio dos mesmos via e-mail.

```
    #!/bin/sh
    PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
    gunzip /var/log/mail.log.0.gz
    pflogsumm /var/log/mail.log.0|formail -c -I"Subject: Estatísticas de e-mail
meudominio.com.br"
    -I"From: pflogsumm@meudominio.com.br" -I"To: meuemail@meudominio.com.br"
    -I"Received: from www.meudominio.com.br ([192.168.0.100])" | sendmail
meuemail@meudominio.com.br
    gzip /var/log/mail.log.0
    exit 0
```

Torne esse arquivo um executável:

```
chmod 755 /usr/local/sbin/postfix_report.sh
```

Após isto, adicione a seguinte linha no `crontab` (use `crontab -e`):

```
0 7 * * * /usr/local/sbin/postfix_report.sh &> /dev/null
```

2.10.10 Análise e desempenho

Existem várias ferramentas para análise de log's, uma delas é o `mailgraph`, para instalar use:

```
apt-get install isoqlog rrdtool mailgraph
```

O `mailgraph` é uma ferramenta simples e prática que usa o `RRDtool` para construir gráficos a partir dos logs do `Postfix`, com incrível riqueza de detalhes. Qualquer administrador de redes que já tenha usado o `MRTG` sabe quanto pode ser útil acompanhar comportamentos e tendências de sistemas por intermédio de gráficos históricos - e o `mailgraph` facilita o processo de acompanhamento do `Postfix` - não apenas da distribuição de mensagens, mas também dos erros, vírus e spam. Caso seja necessário mudar os parâmetros do `mailgraph`, use:

```
dpkg-reconfigure mailgraph
```

Para executar o mailgraph, você deverá proceder como se hospedasse uma nova home page (seção 2.1 página 21), no entanto, será hospedado um script `CGI`.

```
cp -p /usr/lib/cgi-bin/mailgraph.cgi /var/www/www.example.com/cgi-bin
```

Onde, /var/www/www.example.com/ é o local onde será hospedado o site. Para executar, uma vez que o site esteja hospedado use:

```
http://www.example.com/cgi-bin/mailgraph.cgi
```

2.11 Servidor de E-mails

Um servidor de email geralmente é composto de pelo menos duas partes, uma responsável pelo *envio* e outra responsável pelo *recebimento*. Onde esses serviços podem ou não estar na mesma máquina

2.11.1 Envio de Emails (SMTP)

Um dos servidores de e-mail´s mais simples e se utilizar é o Postfix (existem centenas de outros). O primeiro passo é instalar o `postfix`:

```
apt-get install postfix
```

```
apt-get install postfix-ldap
apt-get install postfix-mysql
apt-get install postfix-pgsql
```

O primeiro pacote é responsável especificamente pelo `postfix`, já os outros dois são opcionais, mas permitem facilidades para autenticação. Caso o postfix não seja executado automaticamente após a instalação, execute-o:

```
/etc/init.d/postfix start
```

Importante, durante a instalação o `postfix` fará algumas perguntas básicas sobre o funcionamento do `SMTP`. A configuração básica do `Postfix`, já é funcional.

O `Postfix` é um `MTA`, com bastante robustez, desempenho e maior facilidade na manutenção e configuração. Além disso, `Postfix` é capaz de emular várias funções do `Sendmail`, evitando assim modificações nas aplicações que utilizam o `Sendmail`. Outra característica importante do `Postfix` é a sua construção modular, facilitando a manutenção do código e permitindo a implementação de novas funcionalidades mais facilmente.

Para uma implementação bem-sucedida do `Postfix` é necessário, uma interface de rede instalada e configurada e um servidor `DNS` instalado e configurado.

Caso precise modificar as configurações do `Postfix`, edite o arquivo `/etc/postfix/main.cf`, sempre que modificar o arquivo de configuração, restarte o seriço `/etc/init.d/postfix restart`.

Para testar se o mesmo está funcionando, execute `telnet localhost 25`, para sair digite `quit`.

2.11.2 Recebimento de E-mail´s POP3 e IMAP

Os protocolos `POP3` e `IMAP` são responsáveis pelo transporte de mensagens recebidas do servidor de e-mail para o cliente de e-mail do usuário. O protocolo `POP3` é mais antigo e mais simples, é mais popular e praticamente todos os programas clientes de e-mail o suportam. O protocolo `IMAP` é mais novo e possui mais funções que o `POP3`, no entanto nem todos os programas clientes de e-mail o suportam. O suporte a estes protocolos não é feito pelo `Postfix`, mas sim por outros servidores.

Para implementar um servidor `POP/IMAP` é necessário somente que um servidor de e-mail esteja instalado e configurado. Para instalar o `IMAP` selecione o pacote courier-imap:

```
apt-get install courier-imap
```

Após a instalação, os serviços são automaticamente disparados, caso precise modificar alguma opção, edite o arquivo `/etc/courier/imapd`.

2.11.3 Instalando um WebMail

Existem centenas de web-mails prontos na internet, um dos mais usados é o squirremail, webmiau, entre outros. Vamos apresentar aqui o squirrelmail. Para instalar use:

```
apt-get install squirremail
```

Por padrão o mesmo é instalado na pasta /usr/share/squirremail. Como você pode observar o mesmo não estará habilitado para se apresentar na web. A maneira mais simples é criar um link para os diretórios de disponibilidade (avaiable) e atividade (enable) do Apache (seção 2.1). Você deve ainda adicionar no arquivo de configuração do Apache as seguintes linhas:

```
Alias /webmail/ "/usr/share/squirremail"
DirectoryIndex index.php
```

Para que o webmail funcione é preciso ter instalado também o PHP, pois o mesmo é baseado nessa tecnologia. Após instalado, execute:

```
squirrelmail-configure
```

ou, dentro da pasta /usr/share/squirremail

```
configure
```

Para verificar o funcionamento desse serviço, acesse http://localhost/webmail. Para que todos os usuários tenham uma conta de e-mail, é *fundamental* que seja criado uma pasta `Maildir` dentro do diretório de cada usuário cadastrado. Para criar automaticamente use:

```
maildirmake ~/Maildir
```

Para que a criação seja automática para os novos usuários a serem cadastrados, modifique a estrutura do skeleton, /etc/skel.

```
maildirmake /etc/skel/Maildir
```

2.12 Servidor Simples e Direto de E-mail (Xmail)

O `Xmail` é uma suíte completa, de fácil instalação com ferramentas para agilizar a configuração. O `XMail` é um servidor de e-mail completo, com suporte a `SMTP` e `POP` e permite uma perfeita adequação com o `webmail Uebimiau` (`IMAP`). Outro ponto que merece destaque no `XMail` é que as contas de usuários não precisam ser usuários do sistema, tornando a configuração ainda mais flexível. Para instalar digite:

```
apt-get install xmail
```

Durante a instalação, aceite as respostas padrão, e informe o que sabe ou deseja. Se possuir outro servidor de e-mail sendo executado, ele será desinstalado durante a instalação e o `XMail` inicializado.

Após a instalação são criados os seguintes arquivos de configuração e tabelas de dados.

```
aliasdomain.tab   domains.tab            mailusers.tab    smtpgw.tab
aliases.tab       extaliases.tab         pop3.ipmap.tab   smtp.ipmap.tab
cmd_line          filters.in.tab         pop3links.tab    smtp.ipprop.tab
ctrlaccounts.tab  filters.out.tab        server.tab       smtprelay.tab
ctrl.ipmap.tab    filters.post-data.tab  smtpauth.tab     spam-address.tab
default_domain    filters.pre-data.tab   smtpextauth.tab  spammers.tab
dnsroots          finger.ipmap.tab       smtpfwd.tab      userdef.tab
```

Sempre que desejar reiniciar/iniciar o serviço use (padrão Debian):

```
/etc/init.d/xmail start
```

Todos os arquivos de configuração do `XMail` ficam em */etc/xmail/*. Como o Xmail já possui uma instalação razoavelmente adequada, devemos alterar somente o necessário para uma configuração básica (Importante onde existe a palavra <tab>, use um caracter de tabulação para separar um item do outro e <newline> significa nova linha).

- `ctrlaccounts.tab`: esta é a tabela que contém o login e senha do administrador do servidor. Ex: `"admin" <tab> "enc-passwd".`
- `domains.tab:` esta é a tabela que contém os domínio virtuais. Ex:
  `"localhost"`
  `"seu.servidor.com.br"`
- `mailusers.tab`: esta é a tabela que contém os usuários do servidor, na seguinte ordem: `"host"` <tab> `"usuário"` <tab> `"enc-password"` <tab> `"id da conta"` <tab> `"diretório da conta"` <tab> `"tipo de conta> <newline>`

  Ex:
  ```
  "xmailserver.test"  "xmailuser"    "1d08040c09"            1   "xmailuser"     "U"
  "wolfserver.rede"   "postmaster"   "150a1611080416110017"  2   "postmaster"    "U"
  ```
- `server.tab`: nesta tabela ficam as configurações gerais, a princípio não é necessário alterá-la para que o servidor funcione.

Não se preocupe com o "diretório da conta", o `XMail` o configura automaticamente em `/var/spool/xmail`, aliás ali também ficam os diretórios dos domínios virtuais.

2.12.1 Adicionando usuários e administradores

Para acrescentar usuários de e-mail, é necessário cadastrar os mesmos no programa, esse procedimento é realizando, adicionando os usuários na tabela mailusers.tab (em `/etc/xmail/`).

Como pode ser visto nos exemplos das tabelas acima, existe um item chamado "enc-password", isso significa `encrypted password`, ou seja, senha criptografada, assim sendo, devemos criptografar as senhas antes de adicionar nas tabelas. Para isso use o comando:

```
XMCrypt SenhaCriptografar
```

A saída desse comando é uma `string` criptografada, que será utilizada como senha.

Caso seja necessário mais detalhes, os mesmos podem ser obtidos em `http://www.xmailserver.org`.

2.12.2 Instalando um FrontEnd para Facilitar o manuseio

Apesar do `Xmail` ser simples de se utilizar, existe um `frontend` feito em `php` que torna mais fácil ainda o gerenciamento de contas e configurações do `Xmail`. Para instalar esse script devemos procurar a versão mais nova na internet, procure por `phpxmail`. Você pode tentar encontrar em no site da `sourceforge`.

Uma vez que tenha encontrado, devemos devemos descompactar ele em um diretório de acesso do `apache`, pois ele é um `Web FrontEnd` utilizando o `php` que já deve estar instalado (seção 2.1.4 página 22).

```
unzip phpxmail1.4.zip -d /var/www
```

No caso acima, eu estou no mesmo diretório onde está o arquivo recém baixado, e descompactarei ele para o diretório `/var/www` (padrão do `apache2` (seção 2.1 página 21)). Agora precisamos dar a permissão de escrita para a pasta `/var/www/phpxmail`. Após acesso com seu navegador o endereço `http://localhost/phpxmail`. Na página, localize e clique em `Add new server`. Preencha os dados solicitados como no exemplo abaixo:

```
Server hostname: localhost
Server ip address: 127.0.0.1
Server port: 6017
CTRL account: admin
CTRL password: admin
```

Se tudo correu bem, deve aparecer a seguinte mensagem, caso não funcione, faça o procedimento abaixo e retorne a realizar o anterior:

```
Server localhost was added successfully
```

Caso não seja reconhecido o phtml, você deverá acrescentar essas linhas na configuração do `apache (/etc/apache2/apache.conf)` junto ao agrupamento `<IfModule mod_mime.c>`.

```
AddType application/x-httpd-php .php .phtml
```

Agora precisamos fazer com que o `Xmail` atenda aos comandos do `PHPXmail`. Devemos dizer para o `Xmail` quem é o seu administrador e qual é a sua senha. Estes dados estão gravados em `/var/www/phpxmail/servers.php`.

```
localhost      127.0.0.1     6017          admin  0401080c0b    0
```

Precisamos informar ao `Xmail` através do arquivo `/etc/xmail/ctrlaccounts.tab`, editando-o com um editor preferido. Ele deverá ficar assim ou pelo menos com o mesmo nome de usuário e senha:

```
"admin"  "0401080c0b"
```

Volte agora ao seu navegador e clique em login, em `User name` digite: `admin` e em `Password`: `admin`. Se, de novo, deu tudo certo, deve aparecer a seguinte mensagem:

```
Logged in as Server admin admin on localhost
```

2.12.3 Criando um domínio virtual

Basta dar um clique em server domains e em new domain. Vamos criar um domínio de exemplo, para isto digite em Domain Name: exemplo.com.br, em Postmaster password: admin e finalmente clique em submit. Deverá aparecer a seguinte mensagem:

```
New domain exemplo.com.br created successfully.
New user postmaster@exemplo.com.br created successfully.
```

2.12.4 Criando usuários no domínio:

Clique em `server domains` e selecione o domínio no qual deseja criar o usuário e clique em `new user` e digite em:

```
User name: alexandre
User password: admin
```

Deverá aparecer a mensagem:

```
New user alexandre@exemplo.com.br created successfully.
```

2.12.5 Instalando o UebiMiau

O UebiMiau é um webmail bastante simples de se utilizar, porém com vários recursos interessantes. Para instalar o mesmo devemos procurar na internet a sua última versão (o UebiMiau é baseado em php, assim, possui as mesmas necessidades da instalação do phpxmail). Para instalar pode ser tentado no sourceforge.

Após realizado o download, é necessário descompactar o arquivo no diretório onde o script será armazenado.

```
unzip uebimiau.zip -d /var/www/uebimiau
```

Realizado esse procedimento, devemos dar os devidos direitos ao `UebMiau`. Para acessar o mesmo, entre no browser preferido e digite

```
http://localhost/uebimiau
```

Idealmente você precisará configurar o arquivo `/var/www/uebimiau/config.php` modificando pelo menos:

```
$temporrary_directory = "/tmp/uebimiau/" ;
$smtp_server = "seu.servidor.smtp.com.br";
$quota_limit= 8192; //para 8mb de limite
$pop3_servers[0]["domain"]="dominio.com.br";
$pop3_servers[0]["server"]="seuservidor.dominio.com.br";
$default_language=0;
```

Voce precisará tirar o seu `php` do `safe mode`, para isso edite o arquivo `/etc/php4/apache2/php.ini` ou `php5`:

```
safe_mode = off ;
safe_mode_allowed_env_vars= ;
```

2.12.6 Enviando e-mail's via sendmail

O `sendmail` é um comando juntamente instalado com o `Xmail`, pode ser muito útil, sendo utilizado como teste. Os principais parâmetros usados por ele são:

```
sendmail [-t] [-f...] [-F...] [--input-file fname] [--xinput-file fname] [--rcpt-file fname] [--] recipient ...
```

```
-f{mail from}        de quem...
-F{ext mail from}    para quem...
-t                   extrai os alvos dos cabeçalhos 'To:'/'Cc:'/'Bcc:'
-i                   lê uma seqüência de dados até encontrar 2 <enter's>.
```

Para testar a funcionalidade do `Xmail`, podemos enviar um e-mail da seguinte maneira:

```
sendmail -fxmailuser@smartdomain user1@dom1 user2@dom2 < msg.txt
```

2.13 Servidor DNS

Na internet, os servidores DNS formam uma gigantesca base de dados distribuída, que tem uma função crítica no funcionamento da rede. No topo da cadeia, temos os `root servers`, 14 servidores espalhados pelo mundo que tem como função responder a todas as requisições de resolução de domínio. Na verdade eles não respondem nada, apenas delegam o trabalho para servidores menores, responsáveis individuais dos domínios.

Um nome de domínio é lido da direita para a esquerda. Temos os domínios primários (`top level domains`, ou `TLD's`), como .com, .net, .info, .cc, .biz, e em seguida os domínios secundários (`country code TLD's`), que recebem o prefixo de cada país, como .com.br ou .net.br. Neste caso, o *com* é um subdomínio do domínio *br*.

A *Internic* cuida dos registros dos domínios raiz (.com, .org, .info e outros), enquanto a Fapesp responde pelos domínios com extensão .br (.com.br, .org.br, etc.).

O registro de domínios na *Internic* é menos burocrático, pois você não precisa ter uma empresa registrada. De qualquer forma, registrando seu domínio na Fapesp ou na Internic, você precisará fornecer dois endereços de `DNS`, para onde serão enviadas as consultas referentes ao seu domínio.

É aqui que entra a questão da autoridade. Sempre que é necessário resolver um domínio, o cliente faz uma requisição para o servidor `DNS` da rede, ou do provedor, informado na configuração da rede. O servidor contata um dos `root servers` e pergunta sobre o domínio. Se for um domínio .br, eles encaminham a requisição ao servidor da Fapesp, que é a autoridade responsável. Ele, por sua vez, verifica em sua base de dados quem é o servidor responsável pelo domínio e encaminha novamente a requisição para ele.

Ao registrar um domínio, você passa a ter autoridade sobre ele, e pode criar subdomínios a forma como quiser, como "fulano.meunome.com.br" ou "vendas.minhaempresa.com". Vejunzip phpxmail1.4.zip -d /var/wwwa o caso dos servidores de hospedagem gratuita, como o HPG & cia. que criam milhões de subdomínios para as páginas hospedadas.

Resolver um nome de domínio é uma operação que pode demorar alguns segundos, por isso os servidores `DNS` armazenam um `cache` de domínios já resolvidos, minimizando o número de requisições. É por isso que quando você faz alguma mudança na configuração do domínio, demoram algumas horas para que ela se replique.

É por isso que, às vezes, você não consegue acessar um determinado site usando o `DNS` do provedor (que está desatualizado), mas consegue usando um `DNS` local, ou outro servidor qualquer.

Ao configurar um servidor `web` com o `Apache` (seção 2.1), podemos hospedar vários sites no mesmo servidor usando `virtual host`.

A idéia aqui é que o visitante digita o nome de domínio do site no navegador e o `Apache` se encarrega de enviá-lo ao diretório correto. Mas, para que o cliente chegue até o servidor, faltam mais duas peças importantes.

A primeira é o registro do domínio, que pode ser feito na Fapesp, Internic ou outro órgão responsável. Ao registrar, você precisa fornecer dois endereços de `DNS`. Em muitos casos, o segundo `DNS` não é obrigatório, ele é apenas uma segurança para o caso do primeiro sair fora do ar.

Uma opção muito usada para o segundo `DNS` é pedir para que algum amigo que também possua um servidor dedicado seja seu `DNS` secundário. Ele precisará apenas adicionar a configuração do seu domínio na configuração do `DNS`, o que é rápido e indolor.

Ao alugar um servidor dedicado, é comum que você receba dois ou mais endereços IP's válidos. Originalmente, seu servidor vai estar configurado para usar apenas um deles, mas você pode ativar o segundo.

A partir daí, seu servidor passa a responder pelos dois endereços IP, e você pode usá-lo simultaneamente como `DNS` primário e secundário.

O servidor DNS mais usado no Linux é o Bind, que aprenderemos a configurar aqui. Não existe problema em instalá-lo no mesmo servidor onde foi instalado o Apache e o Proftpd, embora do ponto de vista da segurança o ideal seja utilizar servidores separados.

Para instalar o Bind, procure pelo pacote bind, e instale-o usando:

```
apt-get install bind
```

O arquivo de configuração principal é o /etc/bind/named.conf. Em versões antigas, o arquivo pode ser simplesmente /etc/named.conf. Por padrão, o Bind já vem configurado para trabalhar como um servidor DNS de cache para a rede local. Inicie o serviço com o comando:

```
/etc/init.d/bind start
```

Se a sua máquina servidora já acessava a internet antes, ou seja, já possuía um apontador para um DNS externo, é só apontar o DNS das suas máquinas de sua rede para o seu servidor, que ele automaticamente vai começar a armazenar os DNS. Caso algum cliente da rede precise de um DNS o qual seu servidor não saiba, o mesmo vai acionar o DNS de sua configuração, formando assim um cache de DNS.

Para configurar o Bind para responder pelos seus domínios, você deverá editar o arquivo /etc/bind/named.conf, acrescentando no final do arquivo (supondo DNS primário 200.134.23.9 e secundário 200.134.23.10):

```
zone "server.com.br" IN {
      type master;
      file "/etc/bind/db.server";
};
```

Ao usar um servidor DNS secundário, inclua a linha allow-transfer, especificando o endereço IP do segundo servidor, como em:

```
zone "server.com.br" IN {
      type master;
      file "/etc/bind/db.server";
      allow-transfer {200.134.23.10;};
};
```

Caso queira, é recomendado que você use o arquivo /etc/bind/named.conf.local (que é processado como se fosse parte do /etc/bind/named.conf principal). A existência deste arquivo separado, visa separar a configuração geral do servidor, da configuração dos domínios, minimizando a possibilidade de erros. Mas, na verdade, o efeito de editar qualquer um dos dois arquivos é o mesmo.

Na configuração acima, a opção zone "server.com.br" na primeira linha indica o domínio que estamos configurando, como registrado na Fapesp. Já a opção file "/etc/bind/db.server" especifica o arquivo onde vai a configuração deste domínio.

O responsável pelo domínio "server.com.br" (type master;), sempre que receber uma requisição vai responder de acordo com o especificado no arquivo db.server, configurado em file "/etc/bind/db.server"; e que o servidor *200.134.23.10*, que representa o DNS secundário, pode assumir a responsabilidade sobre o domínio em caso de problemas com o titular, informado pelo opção allow-transfer.

Em seguida você precisa adicionar a configuração do domínio no arquivo /etc/bind/db.server. O conteúdo do arquivo deve ser:

```
@ IN SOA servidor.servidor.com.br. hostmaster.servidor.com.br. (
2006040632 3H 15M 1W 1D )
NS servidor.servidor.com.br.
IN MX 10 servidor.servidor.com.br.
servidor.com.br. A 200.134.23.9
www A 200.134.23.9
ftp A 200.134.23.9
smtp A 200.134.23.9
```

Neste arquivo, a formatação é importante. Você pode usar espaços e tabs para organizar as opções,

mas existem algumas regras. As linhas `IN SOA` até `IN MX` precisam ficar justificadas, e você não pode esquecer dos espaços entre as opções. Caso queira incluir comentários, use ";" ao invés de "#", como em outros arquivos. Vamos então a uma descrição detalhada de cada um dos campos:

```
@ IN SOA servidor.servidor.com.br. hostmaster.servidor.com.br. (
```

A `@` na primeira linha indica a origem do domínio e, ao mesmo tempo, o início da configuração. Ela é sempre usada, assim como num endereço de e-mail. O `IN` é abreviação de internet e o `SOA` de `Start of autority`. Em seguida vem o nome do servidor (que você checa usando o comando `hostname`), seguido do e-mail de contato do administrador.

Importante: hostmaster.servidor.com.br. é um email sem @, ou seja, o email do administrador. Você não pode usar arroba, pois esse símbolo tem a função de informar o início de uma nova configuração. Observe o ponto depois do nome do servidor e do email, o ponto serve para informar o domínio raiz.

A primeira linha termina com um parênteses, que indica o início da configuração do domínio. Temos então: `2006061632 8H 2H 1W 1D` , onde `2006061632` é o valor de sincronismo, que permite que o servidor `DNS` secundário mantenha-se sincronizado com o principal, detectando alterações na configuração. Este número é composto da data da última alteração, nesse caso 2006 06 16, (dia de hoje) e um número de dois dígitos qualquer que você escolhe. Sempre que editar a configuração, ou sempre que configurar um servidor `DNS` a partir de um template qualquer, lembre-se de atualizar a data e mudar os dois dígitos.

Os quatro campos seguintes orientam o servidor `DNS` secundário. O primeiro campo indica o tempo que o servidor aguarda entre as atualizações, nesse caso 8 horas. Caso ele perceba que o servidor principal está fora do ar, ele tenta fazer uma transferência de zona, ou seja, tenta assumir a responsabilidade sob o domínio. Caso a transferência falhe e o servidor principal continue fora do ar, ele aguarda o tempo especificado no segundo campo, nesse caso 2 horas e tenta novamente.

O terceiro campo indica o tempo máximo que ele pode responder pelo domínio, antes que as informações expirem, nesse caso 1 semana, tempo mais do que suficiente para você arrumar o servidor principal. E por último, o tempo mínimo antes de devolver o domínio para o servidor principal quando ele retornar, nesse caso 1 dia.

Estes valores são padrão, por isso não existem muitos motivos para alterá-los. A transferência do domínio para o `DNS` secundária é sempre uma operação demorada, por causa do `cache` feito pelos diversos servidores `DNS` espalhados pelo mundo: demora de um a dois dias até que todos atualizem suas tabelas de endereços. A principal prioridade deve ser evitar que o servidor principal fique indisponível em primeiro lugar. As duas linhas que vem a seguir concluem a seção inicial `NS servidor.servidor.com.br. IN MX 10 servidor.servidor.com.br`. A primeira, diz quem são as máquinas responsáveis pelo domínio. Ao usar apenas um servidor `DNS`, você simplesmente repete o nome do servidor, seguido pelo domínio, como adicionamos na primeira linha. Caso você esteja usando dois servidores, um abaixo do outro.

A linha `IN MX` é necessária sempre que você pretende usar um servidor de `e-mails` (muitas ISP´s chamam o servidor de email de servidor MX). Os números indicam a prioridade de cada servidor. O servidor da primeira linha tem prioridade 10, por isso é o primário. Caso tenha um segundo, poderia colocara prioridade 20 e por isso só assume em casos de problemas com o primário. Usar um segundo servidor de e-mails, num domínio separado, adiciona uma camada extra de redundância e evita que você perca e-mails caso seu servidor fique temporariamente fora do ar.

Depois destas linhas iniciais, temos a parte mais importante, onde você especifica o IP do servidor e pode cadastrar subdomínios. No exemplo foi incluído também três subdomínios, o `www`, `ftp` e `smtp`, ambos relacionados ao IP do servidor. Isso permite que os visitantes digitem *www.servidor.com.br* ou *ftp.servidor.com.br* no navegador. Ao trabalhar com subdomínios, você pode relacioná-los com IP's ou domínios diferentes.

Para testar o seu servidor `DNS`, use:

```
dig servidor.com.br
```

Isso faz com que ele pergunte diretamente ao seu servidor, o que permite testar a configuração imediatamente.

2.13.1 Configurando um DNS para a Intranet

Esta mesma configuração pode ser usada para criar um servidor `DNS` particular, para a sua rede local.

Com isso você poderá acessar todos os micros através de nomes de domínio, como na internet, ao invés de ficar decorando endereços IP.

Você deve começar instalando o `Bind` no servidor da rede. Configure todos os micros da rede interna para utilizarem o IP deste servidor como `DNS` primário e adicione a configuração dos domínios no `Bind`.

Se, por exemplo, os endereços dos três micros são respectivamente 192.168.0.2, 192.168.0.3 e 192.168.0.4 e o servidor é o 192.168.0.1, e o nome do domínio principal é rede, nesse caso adicione no final do arquivo `/etc/bind/named.conf`:

```
zone "rede" IN {
type master;
file "/etc/bind/db.rede";
};
```

Dentro do arquivo `/etc/bind/db.rede`, devemos acrescentar:

```
@ IN SOA servidor.rede. hostmaster.rede. (2006040656 3H 15M 1W 1D)
NS servidor.rede.
rede. A 192.168.0.1
administracao A 192.168.0.2
vendas A 192.168.0.3
contabilidade A 192.168.0.4
```

Temos aqui o *rede*, que é o próprio servidor, além do *administracao.rede*, *vendas.rede* e *contabilidade.rede*, respondendo pelos endereços especificados. Nesse caso não foi incluída a linha `IN MX`, pois como já comentando, ela somente é necessária quando incluir um servidor de e-mails (servidor MX).

2.14 Autenticação Centralizada com LDAP.

`LDAP` significa `Lightweight Directory Access Protocol`, ou seja, Protocolo Leve de Acesso a Diretórios. O `LDAP` é executado em cima do protocolo `TCP/IP` ou outras conexões de transferência de serviços.

Serviço de diretório é um banco de dados otimizado para ler, navegar e procurar. Os diretórios contém descritores formados por atributos e características e um suporte sofisticado para filtros.

As atualizações dos diretórios são simples e rápidas. São feitos para respostas rápidas de um alto volume de operações de buscas. É possível replicar informações largamente para aumentar a disponibilidade dos recursos.

Até algum tempo atrás, o método mais usado para isso era o `nis` com `nfs`, porém de algum tempo para cá, o `LDAP` está sendo mais utilizado, principalmente pela sua capacidade de integração com outros serviços.

2.14.1 Configuração LDAP Servidor

Para instalar o servidor `LDAP`, você deve executar:

```
apt-get install ldap-utils slapd nscd libnss-ldap libpam-ldap libpam-passwdqc
```

Após a instalação será solicitado algumas informações como domínio de ação, nome da organização, senha de administrador entre outras características. Leia e tente responder, caso não saiba, não se preocupe, pois modificaremos os arquivos de configuração manualmente.

Crie um diretório para armazenar os certificados de criptografados. e já fique no mesmo para os próximos procedimentos.

```
mkdir /etc/ldap/certificados
cd /etc/ldap/certificados
```

Gere os certificados, executando um-por-um os comandos abaixo:

```
# Gera certificado servidor
openssl genrsa -des3 -out server.key 4096
openssl rsa -in server.key -out server.key
openssl req -new -key server.key -out server.csr
```

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.csr

# Gera certificado cliente
openssl genrsa -des3 -out client.key 1024
openssl rsa -in client.key -out client.key
openssl req -new -key client.key -out client.csr
openssl x509 -req -days 365 -in client.csr -signkey client.key -out client.csr
```

Após ter gerado os certificados, devemos vincular os mesmos aos mecanismos de autenticação no servidor, para isso edite o arquivo `/etc/ldap/slapd.conf`, e adicione os seguinte comandos:

```
# Opcoes SSL
TLSCertificateFile     /etc/ldap/certificados/server.csr
TLSCertificateKeyFile  /etc/ldap/certificados/server.key
TLSVerifyClient 0
startls=yes
```

Observe que estamos ligando o servidor `slapd` com os certificados já gerados. Agora edite o arquivo `/etc/ldap/ldap.conf` para vínculo dos clientes:

```
# SSL Options
TLS_CERT        /etc/ldap/certificados/client.csr
TLS_KEY         /etc/ldap/certificados/client.key
TLS_REQCERT     allow
```

O arquivo final deve ser algo como:

```
# $OpenLDAP: pkg/ldap/libraries/libldap/ldap.conf,...
BASE    dc=dominio, dc=com
URI     ldap://ldap.dominio.com
#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never
# SSL Options
TLS_CERT        /etc/ldap/certificados/client.csr
TLS_KEY         /etc/ldap/certificados/client.key
TLS_REQCERT     allow
```

2.14.1.1 Ajustes na PAM

Agora precisamos ajustar o sistema de `login` para estar vinculado com o nosso sistema `LDAP`, para isso devemos ajustar o `Linux-PAM` (`Pluggable Authentication Modules for Linux`), para isso devemos editar os arquivos em `/etc/pam.d/`. Primeiramente vamos modificar o arquivo responsável pela forma como será feita a solicitação de `login` (conferência de `login`), `/etc/pam.d/common-account`:

```
# /etc/pam.d/common-account - authorization settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define
# the central access policy for use on the system.  The default is to
# only deny service to users whose accounts are expired in /etc/shadow.
#
#account        required        pam_unix.so

account sufficient pam_unix.so
account sufficient pam_ldap.so
account sufficient pam_permit.so
```

Uma vez informado como será a conferência de `login`, precisamos informar como será feita a conferência do `password`, para isso edite o arquivo `/etc/pam.d/common-password`:

```
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define  the services to be
```

```
#used to change user passwords.  The default is pam_unix

# The "nullok" option allows users to change an empty password, else
# empty passwords are treated as locked accounts.
#
# (Add `md5' after the module name to enable MD5 passwords)
#
# The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option in
# login.defs. Also the "min" and "max" options enforce the length of the
# new password.

#password   required   pam_unix.so nullok obscure min=4 max=8 md5

# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
# (Replaces the `OBSCURE_CHECKS_ENAB', `CRACKLIB_DICTPATH')
#
#password required        pam_cracklib.so retry=3 minlen=6 difok=3
#password required        pam_unix.so use_authtok nullok md5

password sufficient pam_passwdqc.so min=disabled,16,12,8,6 max=256
password sufficient pam_unix.so use_authtok md5
password sufficient pam_ldap.so use_first_pass use_authtok md5
password required pam_deny.so
```

Será necessário configurar ainda ajustar o arquivo `/etc/pam.d/common-auth` para informar como será a autenticação:

```
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
#auth   required        pam_unix.so nullok_secure

auth sufficient pam_unix.so
auth sufficient pam_ldap.so use_first_pass
auth required pam_deny.so
```

Por último precisamos ajustar o arquivo `/etc/pam.d/common-session`, que é responsável pela seção de `login` (tempo de vida do `login`).

```
# /etc/pam.d/common-session - session-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define tasks to be performed
# at the start and end of sessions of *any* kind (both interactive and
# non-interactive).  The default is pam_unix.
#
session required        pam_unix.so
```

2.14.1.2 Configurações de Host e Base

Chegando nos “finalmentes” do servidor. Aqui se encontram algumas daquelas configurações que foram solicitadas durante a instalação. Caso não tenho configurado corretamente, altere o arquivo `/etc/libnss-ldap.conf`:

```
host ldap.intranet.ttc.inf.br
base dc=intranet,dc=ttc,dc=inf,dc=br
```

E por último o arquivo `/etc/pam_ldap.conf`, alterando as mesmas informações do caso anterior.

2.14.2 Configuração LDAP Cliente

Para configurar os clientes, instala-se os mesmos aplicativos do servidor, mas sem o `slapd`. Execute:

```
apt-get install ldap-utils nscd libnss-ldap libpam-ldap
```

Após instalado, configure os mesmos arquivos da mesma maneira, porém sem as opções de `ssl` no `ldap.conf`.

2.14.3 Criando os grupos e usuários

Para cada grupo que você quer cadastrar usuários, você deverá ter no arquivo `ldif` do grupo, uma entrada como a abaixo:

```
cn=NOME_DO_GRUPO,ou=Group,dc=dominio,dc=com
objectClass: posixGroup
objectClass: top
cn: NOME_DO_GRUPO
userPassword: {Crypt}x
gidNumber: GID_DO_GRUPO
memberUid: USUARIO_MEMBRO_DO_GRUPO
memberUid: OUTRO_USUARIO_MEMBRO_DO_GRUPO
```

Importante: pode ser usado o `migrationtools (apt-get install migrationtools)` para gerar o `ldif` com todo o seu arquivo `/etc/group`, assim, você tira as opções indesejadas, os arquivos do `migrationtools` ficam em `/usr/share/migrationtools`.

Uma vez que ajustou os grupos ou os criou, devemos criar os usuários, o que não é uma operação tão trivial. Primeiramente, devemos gerar a senha:

```
slappasswd
```

Esse comando solicitará uma senha e mostrará na tela a mesma criptografada, outra forma que pode agilizar um pouco é executando um conjunto de rotinas, ou seja, armazenar a senha em uma variável e depois incluir no arquivo de senhas o nome do usuário e sua senha, separados por ":". Para isso execute os comandos abaixo, substituindo o NOME_DO_USUARIO pelo nome do usuário desejado.

```
a=`slappasswd`
echo "NOME_DO_USUARIO: $a" >> senhas.ldap
```

Com um arquivo gerado com cada senha para cada usuário, você irá gerar o `ldiff,` para os usuários, e colocar a senha (sem o nome do usuário) na linha `userPassword` do arquivo `usuarios.ldif` (uma entrada para cada usuário).

```
uid=USUARIO,ou=People,dc=dominio,dc=com
uid: USUARIO
cn: DESCRICAO DO USUARIO
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: COLE AQUI A SENHA DO USUARIO
shadowLastChange: 13417
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: UID_DO_USUARIO
gidNumber: GID_DO_USUARIO
homeDirectory: /HOME/DO/USUARIO
```

Acrescentando os `ldifs` no `slapd`. Não adianta criarmos os arquivos para bonito, então vamos vincular eles, para isso:

```
/etc/init.d/slapd stop
slapadd < grupos.ldif
```

```
slapadd < usuarios.ldif
slapindex
/etc/init.d/slapd start
```

2.14.4 Facilitando as coisas com phpldapadmin

Como foi visto anteriormente, o procedimento de cadastro de usuários e grupos não é uma tarefa trivial, para facilitar a vida do administrador foi criada uma ferramenta bastante útil, o `phpldapadmin`. Para instalar use:

```
apt-get install phpldapadmin apache-ssl
```

2.15 Servidor NIS (Network Information Service)

O `NIS` inicialmente era chamado de Yellow Pages, porém esse nome já era patenteado pela *BellSouth*, então o seu nome teve que ser trocado para `NIS`, no entanto os daemons do `NIS` possuem *yp* em seu nome (*ypbind, ypcat, ypserv*).

O `NIS` permite você ter uma base de autenticação centralizada, sem a necessidade de ficar criando os usuários em todas as máquinas, e sim, em um só servidor. O funcionamento do `NIS` é o baseado em *Chamadas de Procedimentos Remotos* (`RPC`).

Para rodar o serviço NIS, é necessário que você crie mapas de informações, para que ele obtenha informações sobre os usuários, o sistema e a rede. Para isso ele lê os seguintes arquivos e gera os respectivos mapas, sendo que todos os mapas que o `NIS` utiliza são armazenados em /var/yp/.:

```
/etc/passwd - gera o mapa passwd.byname
/etc/shadow - os dados desse arquivo são anexados ao mapa passwd.byname
/etc/group - gera o mapa group.byname
```

Existem outros arquivos que o NIS também usa para gerar mapas:

```
/etc/protocols - protocols.bynumber
/etc/services - services.byname
/etc/aliases - mail.aliases
```

Para instalar tanto o cliente quanto o servidor na distribuição, basta executar o seguinte comando:

```
apt-get install nis autofs
```

O processo de instalação vai perguntar o nome do seu domínio NIS, este é o nome que descreve o nome do seu grupo `NIS`, ele não é o `hostname`. Caso queira mudar o nome você pode editar o arquivo `/etc/defaultdomain` (cuidado que o nome é case sensitive).

Outra coisa muito importante que é que devemos cadastrar todas as máquinas pertencentes os nosso domínio `NIS` no `/etc/hosts` ou então criar um `DNS` para localizar as máquinas.

Após realizar a instalação do NIS, você deverá informar o domínio, e logo em seguida a instalação informa que você deverá editar, */etc/nsswitch.conf e/ou /etc/passwd e /etc/group.* Assim que é terminado a instalação de pacotes, o serviço é inicializado. Para inicializar, parar, reinicializar, e outros use:

```
/etc/init.d/nis start
```

Para fazer com que o NIS funcione, é necessário ajustar alguns arquivos de configuração, vamos começar pelo arquivo `/etc/ypserv.conf`. No nosso caso já está ajustado.

O primeiro passo é editar o arquivo /etc/exports, nele deverão ser colocados o diretório do servidor que vai ser compartilhado com a máquina cliente, o número do IP da máquina que estará acessando o servidor e por fim as opções de segurança. Use seu editor de textos preferido e vamos ao exemplo:

```
/home/diretorio 192.168.0.2(rw)
```

Onde, /home/diretorio é o diretório do servidor que vai ser compartilhado com a máquina cliente, o IP 192.168.0.2 é o IP da máquina cliente, e finalmente o (rw) é a permissão dada à maquina cliente, no caso read and write, para mais detalhes veja a seção 2.6.1 página 29, que trata especificamente de NFS. Após

salvar o arquivo, execute o comando:

```
exportfs
```

Com esse comando, seu arquivo de configuração `/etc/exports` será lido e o kernel atualizado com as mudanças realizadas.

2.15.1 Servidor NIS/NFS

Para podermos fazer com que o NIS faça autenticação remotamente, temos que primeiro ter um serviço que permita isso. Basicamente o que você precisa é de editar o arquivo `/etc/exports` e colocar as seguintes linhas:

```
/home/ ip_da_maquina_remota(rw,no_root_squash)
```

O que temos aqui é o seguinte:

- `/home/` - Este é o diretório que será exportado;
- `ip_da_maquina_remota`, este é o ip da máquina que terá acesso ao diretório exportado. Pode-se usar o coringa * para liberar para qualquer máquina.
- `(rw,no_root_squash)`, são as opções que usamos aqui, onde *rw* permite a leitura e gravação para os IPs especificados e *no_root_squash* dá permissão de acesso ao *root* remoto também.

O ypserv.conf normalmente já vem configurado:

```
# Should we do DNS lookups for hosts not found in the hosts table ?
# This option is ignored in the moment.
dns: no
# How many map file handles should be cached ?
files: 30
# Should we register ypserv with SLP ?
slp: no
# xfr requests are only allowed from ports < 1024
xfr_check_port: yes
```

Agora vamos configurar o `/etc/yp.conf`:

```
domain nome_do_dominio hostname x.x.x.x
```

No arquivo `/etc/defaultdomain`, coloque na última linha:

```
NISDOMAIN="nome_do_dominio_nis"
```

onde, `domain nome_do_dominio,` este é o nome do seu domínio NIS, evite colocar o mesmo nome do domínio DNS. Já o `hostname x.x.x.x,` é o ip do seu servidor NIS.

Agora entre no diretório `/var/yp/`, edite o arquivo *Makefile* e troque as seguintes linhas:

```
MERGE_PASSWD=true
MERGE_GROUP=true
```

por:

```
MERGE_PASSWD=false
MERGE_GROUP=false
```

O arquivo securenets:

```
# Always allow access for localhost
255.0.0.0 127.0.0.0
# This line gives access to everybody. PLEASE ADJUST!
x.x.x.x x.x.x.x
```

Para sua segurança, modifique este arquivo de acordo com suas necessidades. No primeiro `x.x.x.x` você coloca sua máscara de rede e no segundo você coloca sua rede para que somente ela tenha acesso. Exemplo:

```
255.255.255.0 192.168.1.0
```

Agora você deve compartililhar os arquivos para que o servidor NIS saiba quais serviços ele poderá buscar na sua rede.

No nosso caso vamos compartilhar os arquivos `/etc/passwd`, `group`, `shadow` e `gshadow`. Para isso vamos colocar um sinal de mais na última linha de cada um deles:

```
echo + >> /etc/passwd
echo + >> /etc/group
echo + >> /etc/shadow
echo + >> /etc/gshadow
```

Agora no servidor NIS, entre no diretório `/var/yp/`, dê o comando `ypserv` e depois `make`. Deverá aparecer uma pasta com o nome do seu domínio. Outra coisa, o diretório home do cliente deve sempre ser o mesmo diretório home do servidor.

2.16 VPN com OpenVPN

`VPN` significa *Virtual Private Network*, essa é uma técnica utilizada para fazer um "túnel" seguro entre duas redes, geralmente separadas pela internet. Os dados são criptografados antes de entrar no túnel e apenas a outra ponta conhece a chave para descriptografar o mesmo, dessa forma, criando um canal de comunicação relativamente seguro. O `OpenVPN` usa criptografia de chaves ao invés de usuário e senha para estabelecer um túnel de `VPN`. Para instalar o `OpenVPN` utilizamos:

```
apt-get install openvpn
```

Após instalar o pacote devemos verificar se o módulo `tun` esteja carregado, se não estiver, use o comando abaixo para poder carregá-lo.

```
modprobe tun
```

Os arquivos de configuração se encontram no diretório /etc/openvpn, assim como o esperado. Entre no diretório de configuração e execute o comando abaixo para gerar uma chave criptografada chamada `vpnkey`.

```
openvpn --genkey --secret vpnkey
```

Para configurar a `VPN`, vamos supor a seguinte configuração de uma matriz e uma filial.

```
Matriz:
eth0: 200.10.10.1
eth1: 192.168.0.1

Filial:
eth0: 230.110.10.34
eth1: 192.168.1.1
```

Para que seja realizada a conexão de forma apropriada, devemos criar o arquivo de configuração da filial na matriz e vice-versa. Assim sendo, no `gateway` da matriz, criamos um arquivo chamado *filial.conf* (nome que desejar, filial1, filial2, ...). Os comandos possuem um pequeno comentário de sua funcionalidade.

```
dev tun                         # Usa o módulo de interface tun já carregado
remote 230.110.10.34            # IP remoto a conectar (No caso o IP da Filial)
ifconfig 10.0.0.1  10.0.0.2     # Primeiro IP local tunelado(matriz) e segundo IP tunelado filial.
secret vpnkey                   # Arquivo que contém a chave.
port 5003                       # Portas UDP para comunicação
ping 15                         # Testa a conexão da VPN
ping-restart 45
ping-timer-rem
persist-tun
persist-key

# Verbosity level.
# 0 — quiet except for fatal errors.
# 1 — mostly quiet, but display non-fatal network errors.
```

```
# 3 – medium output, good for normal operation.
# 9 – verbose, good for troubleshooting
verb 3

# Script's a serem executador quando a VPN for ativada e desativada
up /etc/openvpn/filial.up
down /etc/openvpn/filial.down
```

Para o correto funcionamento do script, se faz necessário os arquivos de `up` e `down`, conforme citados anteriormente no script anterior. Por exemplo, um arquivo *filial.up* poderia ter a seguinte estrutura (criar em `/etc/openvpn`).

```
#!/bin/sh
echo
echo "Criando rotas para redes tuneladas…"
ip route add 192.168.1.0/24 via 10.0.0.2
touch /etc/openvpn/up.running
```

Já o arquivo *filial.down,* que possui a finalidade de eliminar as rotas para a nossa vpn com a filial, poderia ter a seguinte estrutura.

```
#!/bin/sh
echo
echo "Removendo rotas para redes tuneladas…"
ip route del 192.168.1.0/24 via 10.0.0.2
touch /etc/openvpn/down.running
```

Dando continuidade, precisamos criar a situação inversa na filial, ou seja, conectar a matriz. O procedimento inicial é o mesmo anterior, no entanto, ao invés de criar novamente uma nova chave, devemos copiar a chave criada na matriz para a filial, ou seja, copie o arquivo `vpnkey` que você gerou no `gateway` da matriz e coloque dentro da pasta `/etc/openvpn` no `gateway` da filial. Esse procedimento é essencial, pois somente assim os extremos saberão criptografar e descriptografar o conteúdo da comunicação. Como pode ser percebido de ante-mão, esse tipo de criptografia é simétrica, no entanto aumentando um pouco o grau de complexidade da configuração é possível usar chaves assimétricas, aumentando substancialmente a segurança (`openSSL`).

Crie os arquivos abaixo, similares ao da filial, agora com o nome de *matriz.conf* na filial.

```
dev tun                         # Usa o modulo de interface tun carregado anteriormente
remote 200.10.10.1              # IP remoto a conectar (No caso o IP da Matriz)
ifconfig 10.0.0.2 10.0.0.1      # Primeiro IP local tunelado(filial) e segundo IP tunelado matriz.
secret vpnkey                   # Arquivo que contem a chave copiado da matriz.
port 5003                       # Porta UDP, mesma da matriz
ping 15                         # Testa a conexão da VPN
ping-restart 45
ping-timer-rem
persist-tun
persist-key
# Verbosity level.
# 0 – quiet except for fatal errors.
# 1 – mostly quiet, but display non-fatal network errors.
# 3 – medium output, good for normal operation.
# 9 – verbose, good for troubleshooting
verb 3

# Scripts a serem rodados quando a vpn for ativada e quando for desativada
up /etc/openvpn/matriz.up
down /etc/openvpn/matriz.down
```

Para o correto funcionamento do script, se faz necessário os arquivos de `up` e `down`, conforme citados anteriormente no script anterior. Por exemplo, um arquivo matriz.*up* poderia ter a seguinte estrutura (criar em `/etc/openvpn`).

```
#!/bin/sh
echo
echo "Criando rotas para redes tuneladas…"
```

```
ip route add 192.168.0.0/24 via 10.0.0.1
touch /etc/openvpn/up.running
```

Já o arquivo *matriz.down,* que possui a finalidade de eliminar as rotas para a nossa `vpn` com a filial, poderia ter a seguinte estrutura.

```
#!/bin/sh
echo
echo "Removendo rotas para redes tuneladas…"
ip route del 192.168.0.0/24 via 10.0.0.1
touch /etc/openvpn/down.running
```

Para iniciar e para o serviço, tanto nas filiais, quanto na matriz, utilize */etc/init.d/openvnp* seguido de *start, stop, restart*.

2.16.1 VPN Linux x Windows

Para realizar a instalação em uma máquina Windows, devemos instalar o arquivo openvpn-2.0.5-gui-1.0.3-install.exe, ou superior. Após a instalação devemos ajustar o arquivo de configuração de forma similar ao que foi utilizado no Linux.

2.17 Shaper – CBQ (Controle de Banda)

O `shaper` é um controlador de banda baseado no `CBQ` (`Class-Based Queueing`), ou seja, utiliza esquemas baseados em filas e prioridades para controlar a banda de comunicação. O `shaper` na verdade é um script conversor de arquivos de configuração em regras `tc qdisc` baseadas em `CBQ`.

```
apt-get install shaper
```

Uma vez instalado o `shaper`, você pode criar as regras de controle de banda no diretório `/etc/shaper`. Os arquivos de configuração devem obedecer o formato pré-definido: `cbq-(clsid).(nome)`, onde: `(clsid)` é um número hexadecimal de dois-bytes na escala (0002-FFFF), que forma uma classe ID de CBQ e `(nome)` é o nome do *shaper* - (pode ser qualquer texto)

Nos casos em que temos poucos arquivos de configuração, podemos definir o (clsid) como a velocidade do shaper Ex. 64k, cbq-0064.backbone-cliente; 512k, cbq-0512.backbone-cliente. Cada arquivo de configuração deve conter os parâmetros:

```
DEVICE=(int-nome),(banda),(peso)
RATE=(velocidade)
WEITH=(peso/10)
PRIO=(prioridade)
RULE=(ip ou rede a ser controlada)
```

No Parâmetro *DEVICE*, `int-nome:` é o nome da interface a ser controlada ex: eth0, eth1, lo, ppp0, wvlan0, `banda:` é a velocidade do dispositivo. ex: ethernet 10Mbit ou 100Mbit, (peso) é um parâmetro de ajuste que deve ser proporcional a (banda). Como regra teremos que: (peso) = (banda) / 10. Quando houver mais classes não mesma interface, só é necessário especificar o parâmetro (banda) e (peso) uma única vez, conseqüentemente em outros arquivos você poderá ter somente o DEVICE=(nome).

O Parâmetro *RATE,* você informa a banda alocada para a classe. Na limitação de velocidade do shaper, podemos usar: Kbit, Mbit ou bps, Kbps, Mbps como sufixos.

Já o parâmetro *WEIGHT* é um parâmetro de ajuste que deve ser proporcional a (banda). Como regra teremos que: (peso) = (banda) / 10

O parâmetro *PRIO* , informa a prioridade do tráfego para a classe. Quanto mais elevado o número, menor a prioridade. A prioridade 5 é um valor "mágico".

No parâmetro *RULE*, (RULE=[[saddr[/prefix]][:port],][daddr[/prefix]][:port]), Os parâmetros fazem que o filtro u32 controlem o tráfego para cada uma das classes. Você pode usar múltiplas composições de regras por arquivo de configuração. Exemplos:

```
RULE=10.1.1.0/24:80
```
controla o tráfego de rede que passa através da porta 80 na rede 10.1.1.0
```
RULE=10.2.2.5
```

controla o tráfego de rede que passa através de qualquer porta ou do host 10.2.2.5

```
RULE=:25,10.2.2.128/26:5000
```

controla o trafego vindo de qualquer origem, com destino entre a porta 50 a porta 5000 na rede 10.2.2.128

```
RULE=10.5.5.5:80,
```

controla o tráfego vindo da porta 80 do host 10.5.5.5

Uma vírgula **,** colocada após qualquer uma destas regras irá controlar o trafego de saída da rede (`upstream`). Tome cuidado para adicionar a vírgula na interface de rede correta.

Parâmetro *BOUNDED*=yes/no, se definido como *yes* o `shaper` será mantido mesmo que haja banda excedente.

O parâmetro *ISOLATED*=yes/no, se definido como *yes* a banda excedente não será compartilhada.

```
DEVICE=eth0,10Mbit,1Mbit
RATE=128Kbit
WEIGHT=10Kbit
PRIO=5
RULE=192.128.1.0/24
```

Esta configuração diz, que o tráfego na interface *eth0* de *10Mbit* através da rede *192.168.1.0* será processado com a prioridade *5* e uma taxa de shapping de *128kbit.* Note que desta forma somente o tráfego de saída (*upstream*) está sendo controlado.

Imagine que você deseja controlar o tráfego que vem do backbone para o cliente a 28Kbit e o trafego que vai do cliente para o backbone a 128Kbit. Você terá que criar dois arquivos de configuração, um para a eth0 e outro para a eth1.

Arquivo cbq-0028.backbone-client

```
DEVICE=eth1,10Mbit,1Mbit
RATE=28Kbit
WEIGHT=2Kbit
PRIO=5
RULE=192.168.1.0/24
```

Arquivo cbq-0128.client-backbone

```
DEVICE=eth0,10Mbit,1Mbit
RATE=128Kbit
WEIGHT=10Kbit
PRIO=5
RULE=200.1.1.1,
```

Para ter efeito as regras que foram criadas, você deverá executar `/etc/init.d/shaper`, onde o comando lhe apresentará todas as suas possibilidades de uso.

Digitando `/etc/init.d/shaper start`, os arquivos de configuração serão carregados, neste momento sua rede já deve estar sofrendo a ação do shaper. Para testar, basta mudar os parâmetros de configuração e digitar `/etc/init.d/shaper restart`, isto forçara a reinicialização do shaper e as novas configurações serão lidas.

Eventualmente é necessário controlar o `upload`, no entanto o IP da máquina já sofreu NAT, e não é possível saber qual IP controlar, a solução nesses casos é a marcação de pacotes. Para isso, devemos incluir nas regras de `iptables` uma regra para marcar os pacotes oriundos da rede ou IP que desejamos controlar.

```
iptables -A PREROUTING -t mangle -i eth0 -s 192.168.0.0/24 -j MARK –set-mark 3
```

Crie no /etc/shaper/ uma regra cbq-ID.nome, por exemplo, *cbq-0008.192_168_0_0-upload* contendo:

```
DEVICE=eth1,100Mbit,10Mbit
RATE=256Kbit
WEIGHT=26Mbit
PRIO=5
BOUNDED=yes
MARK=3
```

Nesse caso, a rede fica marcado com o número 3, logo após digo em uma regra de CBQ, que o tráfego

será controlado de acordo com a marcação. Outro exemplo de limitação de uma única máquina:

Arquivo cbq-0002.estacao9-in

```
DEVICE=eth1,100Mbit,10Mbit
RATE=32Kbit
WEIGHT=3Kbit
PRIO=5
RULE=192.168.0.2/32
BOUNDED=yes
ISOLATED=yes
```

3 IPTables

O IPTABLES é um software usado para analisar os pacotes que passam entre redes. A partir desse princípio podemos aceitar, rejeitar ou descartar esses pacotes. Através de métodos de controle de acesso baseado em registros em tabelas presentes no KERNEL, um FIREWALL pode monitorar e registrar qualquer informação que estão sendo transmitidas entre as redes locais e externas em tempo real. No Linux, a filtragem de pacotes é implementada no kernel, como modulo ou compilado diretamente através do NETFILTER. Geralmente o iptables já vem instalado com os pacotes padrão da distribuição, caso o iptables não esteja instalado, você pode instalar ele com o seguinte comando:

```
apt-get install iptables
```

O iptables possui 3 chains básicas INPUT, OUTPUT e FORWARD as quais não podem ser apagadas, para gerenciar essas operações, temos as seguintes opções:

```
Criar nova chain (-N);
Apagar uma chain vazia (-X);
Mudar a política de uma chain built-in (-P);
Listar as regras de uma chain (-L);
Apagar todas as regras de uma chain (-F);
Zerar os contadores de pacotes e bytes de todas as regras de uma chain (-Z);
```

Há várias formas de manipular regras dentro de uma chain:

```
Adicionar uma nova regra na chain (-A);
Inserir uma nova regra em alguma posição da chain (-I);
Substituir uma regra em alguma posição da chain (-R);
Apagar uma regra em alguma posição da chain (-D);
Apagar a primeira regra que associa (com alguma condição) numa chain (-D);
Apagar todas as regras de um chain ou em todas as chain (-F)
```

O iptables pode ser um módulo chamado iptable_filter.o, que deve ser automaticamente carregado quando você rodar iptables pela primeira vez. Ele também pode ser compilado diretamente no kernel, como já comentado.

Antes dos comandos do iptables rodarem, não haverão regras em quaisquer chains INPUT, FORWARD ou OUTPUT, e todas as chains terão a política ACCEPT. A política padrão da chain FORWARD pode ser alterada fornecendo o parâmetro forward=0 para o módulo iptable_filter.

3.1 Operações em uma única regra

Usualmente, você provavelmente utilizará os comandos para adicionar (-A) e apagar (-D). Os outros (-I para inserir e -R para substituir) são apenas extensões destes conceitos. Cada regra especifica uma série de condições que o pacote deve atender, e o que fazer com o mesmo se ele atendê-las *um alvo target*.

Por exemplo, você pode desejar descartar todos os pacotes ICMP vindos do endereço IP 127.0.0.1. Então neste caso nossas condições são que o protocolo precisa ser ICMP e o endereço de origem deve ser 127.0.0.1. Nosso alvo target é DROP. O IP 127.0.0.1 é a interface *loopback*, que existirá mesmo que você não tenha nenhuma conexão de rede real. Pode-se utilizar o programa *ping* para gerar esse tipo de pacote (ele simplesmente manda um ICMP tipo 8 (requisição de eco) que é respondido pelos hosts com um ICMP tipo 0 (resposta de eco)).

```
ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.2 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
```

Acrescentando a regra, o resultado fica:

```
iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
```

```
ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
```

Pode-se ver que o primeiro ping é bem sucedido (a opção `-c 1' diz ao ping para mandar um único pacote). Então foi adicionada (**-A**) à chain `INPUT`, uma regra especificando que pacotes vindos de 127.0.0.1 `-s 127.0.0.1` com o protocolo `ICMP` *-p icmp* devem ser mandados para `DROP` ***-j*** *DROP*.

Logo depois, testamos nossa regra, usando um segundo ping. Haverá uma pausa antes que o programa desista de esperar pelo pacote que nunca viria.

3.1.1 Apagando Regras

Pode-se apagar regras de duas maneiras. Primeiramente, desde que sabemos que existe apenas uma regra na chain `INPUT`, podemos utilizar um número para designar a regra, como abaixo:

```
iptables -D INPUT 1
```

Para apagar a regra número 1 na chain `INPUT`. A segunda forma, é fazer o mesmo que faríamos para adicionar a regra, trocando **-A** por **-D**. Isso é útil quando você tem uma chain muito complexa e não quer contar para descobrir o número da regra. Neste caso usaria-se:

```
iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP
```

A sintaxe do **-D** deve ter exatamente as mesmas opções que seriam passadas para a opção **-A** (ou **-I** ou **-R**). Se existem várias regras idênticas na mesma chain, apenas a primeira será apagada.

3.1.2 Especificações para filtragem

A opção **-p**, serve para especificar o protocolo, e **-s** para especificar o endereço de origem, mas existem outras opções que podem ser utilizadas para especificar outras características dos pacotes. O que segue é uma explicação exaustiva. Especificando endreços IP de origem e destino

Endereços IP de Origem (**-s**, **--source** ou **--src**) e destino (**-d**, **--destination** ou **--dst**) podem ser especificados em quatro formas diferentes. A mais comum é utilizar o nome completo, como *localhost* ou o `DNS`. A segunda é dizer o IP, como *127.0.0.1*.

A terceira e a quarta formas permitem a especificação de um grupo de endereços IP, como *199.95.207.0/24* ou *199.95.207.0/255.255.255.0*. Ambas especificam qualquer endereço IP de *199.95.207.0* até *199.95.207.255*. Os dígitos depois da / dizem quais partes do endereço IP são significantes. /32 ou /255.255.255.255 é o padrão (associando o endereço IP inteiro). Para especificar qualquer endereço IP pode-se utilizar /0, conforme descrito abaixo:

```
iptables -A INPUT -s 0/0 -j DROP
```

Isso raramente é utilizado, uma vez que o efeito da regra acima é o mesmo que se não fosse especificada nenhuma origem.

3.1.3 Especificando Inversão

Muitas flags, incluindo **-s** (**--source**) e **-d** (**--destination**) podem ter seus argumentos precedidos de **!** para associar-se com endereços *diferentes* aos passados à opção. Por exemplo, **-s !** *localhost* associa-se com qualquer pacote que não venha de `localhost`.

3.1.4 Especificando protocolo

O protocolo pode ser especificado com **-p** (**--protocol**). Os protocolos podem ser `TCP`, `UDP` or `ICMP`, existe ainda a cláusula `ALL`. Digitar em maiúsculas ou minúsculas não faz diferença, *tcp* funciona tão bem como *TCP*.

Se o nome do protocolo é precedido de **!**, a fim de invertê-lo, como **-p!** `TCP`, a regra especificará todos os pacotes que não são `TCP`.

3.1.5 Especificando uma interface

As opções **-i** (**--in-interface**) e -o (**--out-interface**) especificam o nome de uma interface a especificar. Uma interface é um dispositivo físico pelo qual o pacote veio **-i** ou está saindo **-o**. Pode-se usar o comando `ifconfig` para listar as interfaces ativas.

Pacotes atravessando a chain `INPUT` não possuem interface de saída, logo qualquer regra utilizando **-o** não terá sentido ou função. Da mesma forma, pacotes atravessando a chain `OUTPUT` não têm interface de entrada, logo qualquer regra utilizando `-i` não tem sentido. Apenas pacotes passando pela chain `FORWARD` têm interfaces de entrada e saída.

Não há nenhum problema em especificar uma interface que ainda não existe, a regra não associar-se-á com quaisquer pacotes até que a interface torne-se ativa. Isso é extremamente útil para links discados PPP (usualmente com a interface ppp0) e similares.

Como um caso especial, um nome de interface terminando com um + vai associar-se com todas as interfaces (existam elas ou não) que comecem com aquela string. Por exemplo, para especificar uma regra que se associa com todas as interfaces PPP, a opção **-i** ppp+ deve ser criada.

3.1.6 Especificando fragmentos

Às vezes um pacote é muito grande para ser enviado todo de uma vez. Quando isso ocorre, o pacote é dividido em fragmentos, e é enviado como múltiplos pacotes. Quem o recebe, remonta os fragmentos reconstruindo o pacote.

O problema com os fragmentos é que o fragmento incial tem os campos de cabeçalho completos (`IP` + `TCP`, `UDP` e `ICMP`) para examinar, mas os pacotes sub-sequentes só têm um pequeno grupo de cabeçalhos (somente IP, sem os campos dos outros protocolos). Logo examinar o interior dos fragmentos sub-sequentes em busca de cabeçalhos de outros protocolos (como extensões de TCP, UDP e ICMP) é impossível.

Se você está fazendo acompanhamento de conexões ou `NAT`, todos os fragmentos serão remontados antes de atingirem o código do filtro de pacotes, então você não precisa se preocupar sobre os fragmentos.

Caso contrário, é importante entender como os fragmentos são tratados pelas regras de filtragem. Qualquer regra de filtragem que requisitar informações que não existem não será válida. Isso significa que o primeiro fragmento é tratado como um pacote qualquer. O segundo e os seguintes não serão. Então uma regra **-p** `TCP` **--sport** *www* (especificando www como porta de origem) nunca se associar-se-á com um fragmento (exceto o primeiro). O mesmo se aplica à regra oposta **-p** TCP **--sport** **!** *www*.

De qualquer forma, você pode especificar uma regra especial para o segundo e os fragmentos que o sucedem, utilizando a flag **-f** (**--fragment**). Também é interessante especificar uma regra que não se aplica ao segundo e os outros fragmentos, precedendo a opção **-f** com **!**.

Geralmente é reconhecido como seguro deixar o segundo fragmento e os seguintes passarem, uma vez que o primeiro fragmento será filtrado, logo isso vai evitar que o pacote todo seja remontado na máquina destinatária. De qualquer forma, há bugs que levam à derrubada de uma máquina através do envio de fragmentos. A decisão é sua.

Nota para os administradores de rede: pacotes mal formados (pacotes TCP, UDP e ICMP muito pequenos para que o código do firewall possa ler as portas ou o código e tipo dos pacotes ICMP) são descartados quando ocorrem tais análises. Então os fragmentos TCP começam na posição 8.

Como um exemplo, a regra a seguir vai descartar quaisquer fragmentos destinados ao endereço 192.168.1.1:

```
iptables -A OUTPUT -f -d 192.168.1.1 -j DROP
```

3.1.7 Extensões ao iptables: Novas Implementações

O `iptables` é extensível, assim, tanto o kernel quanto o iptables podem ser extendidos para fornecer novas funcionalidades. Algumas dessas extensões são padronizadas, e outras são mais exóticas. Extensões podem ser feitas por qualquer um e distribuídas separadamente para usuários de nichos mais específicos.

As extensões do kernel geralmente estão no subdiretório de módulos do kernel como, por exemplo, `/lib/modules/2.3.15/net`. Elas são carregadas por demanda se seu kernel foi compilado com a opção `CONFIG_KMOD` marcada, não havendo a necessidade de inserí-las manualmente.

As extensões do `iptables` são bibliotecas compartilhadas as quais geralmente estão em `/usr/local/lib/iptables/`, mas uma distribuição os colocaria em `/lib/iptables` ou

`/usr/lib/iptables`.

Extensões vêm em dois tipos diferentes: novos alvos (targets), e novas associações (depois falaremos mais sobre novos alvos `targets`). Alguns protocolos automaticamente oferecem novos testes: atualmente são eles TCP, UDP e ICMP como mostrado abaixo.

Para esses protocolos você poderá especificar novos testes na linha de comando depois da opção **-p**, que carregará a extensão. Para novos testes explícitos, utilize a opção **-m** para carregar a extensão, depois de **-m**, todas as opções da extensão estarão disponíveis.

Para conseguir ajuda com uma extensão, utilize a opção que a carrega **(-p**, **-j** ou **-m**) sucedida por **-h** ou **--help**, conforme o exemplo:

```
iptables -p tcp --help
```

3.2 Extensões TCP

As extensões TCP são automaticamente carregadas se é especificada a opção **-p** *tcp*. Elas provem as seguintes opções (nenhuma associa-se com fragmentos).

--tcp-flags seguida por uma opcional **!**, e por duas strings indicando flags, permite que sejam filtradas flags *TCP* específicas. A primeira string de flags é a máscara, uma lista de flags que serão examinadas. A segunda string diz quais flags devem estar ativas para que a regra se aplique. Por exemplo:

```
iptables -A INPUT --protocol tcp --tcp-flags ALL SYN,ACK -j DROP
```

Essa regra indica que todas as flags devem ser examinadas (`ALL` é sinônimo de `SYN,ACK,FIN,RST,URG,PSH`), mas apenas `SYN` e `ACK` devem estar ativadas. Também há um argumento `NONE` que significa nenhuma flag.

--syn opcionalmente precedido por **!**, é um atalho para **--tcp-flags SYN,RST,ACK SYN**.

--source-port seguido por ! opcional, e uma única porta TCP, ou um conjunto (range) de portas. As portas podem ser descritas pelo nome, conforme listado em `/etc/services`, ou pelo *número*. Conjuntos (ranges) são dois nomes de portas separados por **:**, ou (para especificar uma porta maior ou igual à especificada) uma porta sucedida por **:**, ou (para especificar uma porta menor ou igual à especificada), uma porta precedida por **:**.

--sport é sinônimo de **--source-port**.

--destination-port ou **--dport** funcionam de forma idêntica a *--source-port*, a única diferença é que elas indicam a porta de destino, e não a porta de origem.

--tcp-option seguida por **!** opcional e um número, associa-se com um pacote com a opção TCP igual ao do número passado. Um pacote que não tem um cabeçalho TCP completo é automaticamente descartado se há uma tentativa de examinar suas opções TCP.

3.2.1 Uma explicação sobre as flags TCP

Às vezes é útil permitir conexões TCP em uma única direção, mas não nas duas. Por exemplo, você permitirá conexões em um servidor WWW externo, mas não conexões vindas deste servidor.

Uma tentativa ingênua seria bloquear pacotes TCP vindos do servidor. Infelizmente, conexões TCP necessitam de pacotes bi-direcionais para um funcionamento perfeito.

A solução é bloquear apenas os pacotes utilizados para *requisitar uma conexão*. Tais pacotes são chamados pacotes `SYN` (tecnicamente eles são pacotes com a flag `SYN` marcada, e as flags `RST` e `ACK` desmarcadas, mas dizemos pacotes `SYN` como atalho). Ao não permitir tais pacotes, nós impedimos conexões vindas do servidor.

A opção **--syn** é utilizada para isso, só é válida para regras que especificam TCP como protocolo. Por exemplo, para especificar conexões TCP vindas do endereço 192.168.1.1: **-p** *TCP* **-s** *192.168.1.1* **--syn**

3.2.2 Extensões UDP

Essas extensões são automaticamente carregadas se a opção **-p** *udp* é especificada. Ela provê as opções **--source-port**, **--sport**, **--destination-port** ou **--dport** conforme detalhado para o TCP acima.

3.2.3 Extensões ICMP

Essas extensões são automaticamente carregadas se a opção **-p** *icmp* é especificada. Ela só possui uma opção diferente das demais:

--icmp-type seguida por **!** opcional, e um nome de tipo *icmp* (por exemplo, `host-unreachable`), ou um tipo numérico (exemplo 3), ou um tipo numérico e código separados por / (exemplo 3/3). Uma lista de tipos *icmp* é passada utilizando-se **-p** *icmp* **--help**.

3.2.4 Outras extensões

Outras extensões no pacote `netfilter` são extensões de demonstração, que (caso instaladas) podem ser chamadas com a opção **-m**.

Mac, Este módulo deve ser explicitamente especificado com **-m** mac ou **--match** mac. Ele é usado para associar-se com o endereço Ethernet (MAC) de origem do pacote, e logo só é útil nas chains `PREROUTING` e `INPUT`. Só provê uma única opção –**mac-source** seguida por **!** opcional e um endereço ethernet passado em notação hexadecima, exemplo **--mac-source** *00:60:08:91:CC:B7*.

Limit, Este módulo deve ser explicitamente especificado com **-m** *limit* ou **--match** *limit.* É usado para restringir a taxa de pacotes, e para suprimir mensagens de log. Vai fazer com que a regra seja válida apenas um número de vezes por segundo (por padrão 3 vezes por hora, com um limite máximo def 5). Possui dois argumentos opcionais:

--limit seguido de um número, especifica a média máxima de pacotes (ou LOGs, etc) permitida por segundo. Pode-se especificar a unidade de tempo, usando **/second**, **/minute**, **/hour** ou **/day**, ou abreviações dos mesmos (assim, *5/second* é o mesmo que *5/s*).

--limit-burst seguido de um número, indicando o máximo de entradas antes do limite tornar-se válido.

Essa extensão pode ser usada com o alvo (target) LOG para criar registros (logs) limitados por taxa de incidência. Para entender o funcionamento disso, olhe a regra abaixo, que loga pacotes com os parâmetros padrão de limite:

```
iptables -A FORWARD -m limit -j LOG
```

Na primeira vez que essa regra é analisada, o pacote será logado, na realidade, uma vez que o padrão máximo é 5, os cinco primeiros pacotes serão logados. Depois disso, passar-se-ão vinte minutos antes de que essa regra faça um novo logo, independente do número de pacotes que entrarem. Além disso, a cada vinte minutos que se passam sem que um pacote associe-se com a regra, o contador é diminuído em uma unidade. Logo, se nenhum pacote associar-se com a regra em 100 minutos, o limite estará zerado, voltando ao estágio inicial.

Nota: não é possível criar uma regra com tempo de recarga superior a 59 horas, então se você configura uma taxa média de um por dia, seu limite máximo deve ser menor que 3. Esse módulo também pode ser usado para evitar uma série de ataques do tipo negativa de serviço (`denial of service - DoS`) com uma taxa mais rápida, a fim de aumentar a sensibilidade do sistema.

Como polítca de proteção, ou você não permite ping, ou então usa uma política de segurança ao receber os ping, dentro dos principais ataques, citamos:

```
iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT
iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j
ACCEPT
iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
```

As regras acima são respectivamente para proteção contra Syn-flood, Port Scanner suspeito, e Ping da morte. Observe que na primeira regra só são aceitos e passados adiante as requisições de conexão feitas com um intervalo de tempo superior à 1 por segundo. A segunda regra, permite realizar uma tentativa de conexão por no máximo uma por segundo. A última somente permite uma resposta de ping no máximo de 1 por segundo.

owner Esse módulo tenta associar-se com várias características do criador do pacote, para pacotes gerados localmente. Só é válido na chain `OUTPUT`, e mesmo assim alguns pacotes (como respostas de ping `ICMP`) podem nao ter dono, e nunca serão analisados pela regra.

--uid-owner userid Associa-se com o pacote se este foi criado por um processo com o *userid* igual ao passado na opção.

--gid-owner groupid Associa-se com o pacote se este foi criado por um processo com o *groupid* igual

ao passado na opção.

--pid-owner processid Associa-se com o pacote se este foi criado por um processo com o *proccessid* igual ao passado na opção.

--sid-owner sessionid Associa-se com o pacote se este foi criado por um processo com o sessionid igual ao passado na opção.

unclean Esse módulo experimental deve ser explicitamente especificado com **-m** *unclean* ou **--match** *unclean*. Ele faz diversas checagens de sanidade nos pacotes.

3.3 Checagens de estado dos pacotes (state match)

O critério para filtragem de pacotes mais útil é provido pela extensão *state* que interpreta a análise do controle da conexão feita pelo módulo `ip_conntrack`. Essa extensão é altamente recomendada.

Especificando **-m** *state* a opção **--state** torna-se disponível, a qual é uma lista dos estados possíveis dos pacotes separada por vírgula . Os estados são:

NEW Um pacote que cria uma nova conexão.

ESTABLISHED Um pacote que pertence a uma conexão existente (um pacote de resposta, um pacote saindo por uma conexão na qual já houveram respostas).

RELATED Um pacote relacionado, mas que não faz parte, de uma conexão existente, como um erro ICMP, ou (com o módulo FTP carregado), um pacote estabelecendo uma conexão de dados FTP.

INVALID Um pacote que não pôde ser identificado por alguma razão: isso inclui falta de memória e erros ICMP que não correspondem a qualquer conexão conhecida. Geralmente tais pacotes devem ser descartados.

3.4 Especificações de alvo (Target)

Agora que sabemos quais as análises podem ser feitas em um pacotes, precisamos de uma forma de dizer o que fazer com os pacotes que passam nas condições que estabelecemos. Isso é chamado de alvo (target) da regra.

Há dois alvos bem simples `DROP` (descartar) e `ACCEPT` (aceitar). Se a regra se associa com o pacote e seu alvo é um desses dois, nenhuma outra regra é consultada: *o destino do pacote já foi decidido*.

Há dois tipos de alvos diferentes dos descritos acima, as extensões e as chains definidas por usuários.

3.4.1 Chains definidas por usuários

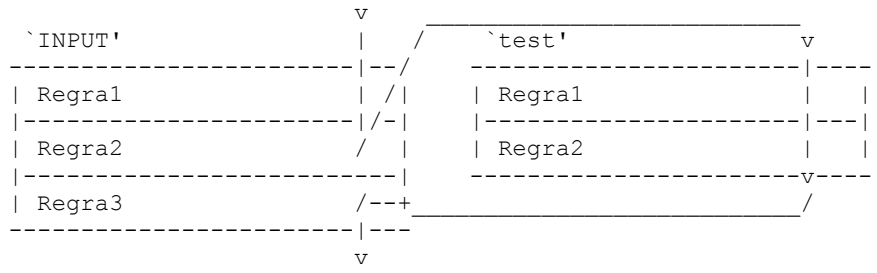
Uma funcionalidade que o iptables herdou do ipchains é a possibilidade do usuário criar novas chains, além das três padrão (`INPUT`, `FORWARD` e `OUTPUT`). Por convenção, chains definidas pelo usuário são escritas em minúsculas, diferenciando-as (como criar chains definidas pelo usuário será descrito abaixo Operações em uma chain).

Quando um pacote associa-se com uma regra cujo alvo (target) é uma chain definida pelo usuário, o pacote passa a ser analisado pelas regras dessa chain definida pelo usuário. Se a chain não decide o que deve ser feito com o pacote, o mesmo volta a ser analisado pela chain padrão que continha a regra que o levava para a chain definida pelo usuário. Considere duas chains: INPUT (a chain padrão) e test (uma chain definida pelo usuário).

```
INPUT                          test
----------------------------   ----------------------------
| Regra1: -p ICMP -j DROP  |   | Regra1: -s 192.168.1.1   |
|--------------------------|   |--------------------------|
| Regra2: -p TCP -j test   |   | Regra2: -d 192.168.1.1   |
|--------------------------|   ----------------------------
| Regra3: -p UDP -j DROP   |
----------------------------
```

Considere um pacote TCP vindo de 192.168.1.1, indo para 1.2.3.4. Ele entra na chain `INPUT` e é testado pela Regra1 - não se associa. Já a Regra2 se associa, e seu alvo é test, logo a próxima regra examinada está no início de test. A Regra1 na chain test se associa, mas não especifica um alvo, então a próxima regra é examinada, Regra2. Ela não se associa, e nós chegamos no final da chain. Então, a chain `INPUT` volta a ser examinada, e como a última regra desta chain que foi examinada foi a Regra2, a regra a ser

examinada agora é a Regra3, que também não se associa com o pacote. Logo, o caminho que o pacote faz é o seguinte:



Chains definidas por usuário podem ter como alvo outras chains definidas por usuário (mas não faça loops, seus pacotes serão descartados se entrarem em loop).

3.4.2 Extensões ao iptables: Novos alvos (targets).

O outro tipo de alvo é uma extensão. Uma extensão-alvo consiste em um módulo do kernel, e uma extensão opcional ao `iptables` para prover opções de linha de comando. Há diversas extensões na distribuição padrão do `netfilter`:

LOG Esse módulo provê logging dos pacotes submetidos. Possui as seguintes opções adicionais

--log-level Seguido de um número de nível ou nome. Os nome válidos (sensíveis a maiúsculas/minúsculas) são **debug**, **info**, **notice**, **warning**, **err**, **crit**, **alert** e **emerg**, correspondendo a números de 7 até 0. Veja a página de manual do `syslog.conf` para uma explicação mais detalhada desses níveis.

--log-prefix Seguido de uma string de até 29 caracteres, esta será adicionada no início da mensagem de log, permitindo melhor identificação da mesma. Esse módulo é útil após de uma comparação por limite (limit match), assim, você não enche seus logs.

REJECT Esse módulo tem o mesmo efeito de `DROP`, a não ser que o remetente tenha enviado uma mensagem de erro `ICMP` *port unreachable* (porta inalcançável). A mensagem de erro `ICMP` não será enviada se (veja RFC 1122):

- O pacote que está sendo filtrado era uma mensagem de erro ICMP no início, ou um tipo desconhecido de ICMP;
- O pacote sendo filtrado era um fragmente sem cabeçalho;
- Já enviamos muitas mensagens de erro ICMP para o mesmo destinatário recentemente.

O **REJECT** também tem o argumento opcional **-reject-with** que altera o pacote de resposta utilizado, veja a página de manual.

3.4.3 Alvos especiais padrão

Há dois tipos especiais de alvos padrão: `RETURN` e `QUEUE`.

RETURN tem o mesmo efeito de chegar ao final da chain: para uma regra numa chain padrão, a política da chain é executada. Para uma chain definida por usuário, a travessia continua na chain anterior, exatamente após a regra que saltou para a chain definida pelo usuário.

QUEUE é um alvo especial, que manda o pacote para uma fila para processamento em nível de usuário. Para isso ser útil, dois componentes são necessários:

- um gerenciador de filas, que trata com a mecânica de passar os pacotes do kernel para o espaço do usuário;
- uma aplicação no espaço do usuário para receber, possivelmente manipular e decidir o que fazer com os pacotes.

O gerenciador de filas padrão para o `iptables` IPv4 `iptables` é o módulo `ip_queue`, que é distribuído com o kernel e é atualmente experimental.

O seguinte é um rápido exemplo de como utilizar o `iptables` para enfileirar pacotes para processamento em nível de usuário:

```
modprobe iptable_filter
modprobe ip_queue
iptables -A OUTPUT -p icmp -j QUEUE
```

Com essa regra, pacotes ICMP de saída gerados localmente (como, por exemplo, criados pelo ping) são passados ao módulo ip_queue, que então tenta mandar os pacotes para uma aplicação em nível de usuário. Se não há nenhuma aplicação em nível de usuário está esperando, os pacotes são descartados.

Para escrever uma aplicação em nível de usuário, utilize a API libipq, que é distribuída com o iptables. Códigos de exemplo podem ser encontradas com as ferramentas testsuite (por exemplo redirect.c) no CVS.

O estado de ip_queue pode ser consultado através de:

```
/proc/net/ip_queue
```

O comprimento máximo da fila (exemplo, o número de pacotes entregues ao espaço do usuário sem que nenhuma resposta fosse passada) pode ser controlado por:

```
/proc/sys/net/ipv4/ip_queue_maxlen
```

O valor padrão para o comprimento máximo da fila é 1024. Uma vez que esse limite foi atingido, novos pacotes serão descartados até que o tamanho da fila diminua abaixo do limite. Bons protocolos como o TCP interpretam pacotes descartados como congestionamento, que talvez voltem quando a fila diminua. De qualquer forma, experimentos para determinar o tamanho ideal da fila pois às vezes o tamanho padrão é muito pequeno.

3.5 Operações em uma chain

Uma funcionalidade muito importante do iptables é a habilidade de juntar regras em chains (cadeias). Você pode dar o nome que quiser para as chains, mas é recomendada a utilização de minúsculas para evitar confusão com as chains padrão ou com os alvos (targets). Nomes de chains podem ter até 31 letras.

3.5.1 Operações com chains

Para criar uma nova chain, utilize as opções **-N** ou **--new-chain**:

```
iptables -N test
```

Para a apagar uma Chain, utilize as opções **-X** ou –**delete-chain:**

```
iptables -X test
```

Há uma série de restrições ao se apagar uma chain, ela deve estar vazia (veja Esvaziando uma chain, logo abaixo) e ela não deve ser alvo de NENHUMA regra. É impossível apagar nenhuma das três chains padrão. Se uma chain não for especificada, *todas* as chains definidas por usuário serão apagadas, desde que seja possível.

Esvaziando uma chain. Há uma forma muito simples de retirar todas as regras de uma chain, utilizando as opções **-F** ou –**flush**. Se uma chain não for especificada, todas as chains serão esvaziadas

```
iptables -F FORWARD
```

Listando uma chain. Pode-se listar todas as regras de uma chain utilizando as opções **-L** ou –**list.** O valor refcnt listado para cada chain definida por usuário é o número de regras que têm tal chain como alvo (target). Este valor deve ser zero (e a chain deve estar vazia) antes que essa chain possa ser apagada. Se o nome da chain é omitido, todas as chains são listadas, até as vazias. Há três opções que podem acompanhar **-L**. A opção **-n** (numérico) é muito útil uma vez que previne que o iptables tente resolver os endereços IP, o que (se você utiliza DNS assim como a maioria das pessoas) causaria grandes atrasos se o DNS não está configurado corretamente, ou você está filtrando requisições DNS. Essa opção também faz as portas TCP e UDP serem impressas como números em vez de nomes.

A opção **-v** mostra todos os detalhes das regras, como os contadores de pacotes e bytes, as comparações *TOS*, e as interfaces. Caso tal opção não seja passada, esses valores sao omitidos.

Note que os contadores de pacotes e bytes são impressos com os sufixos *K*, *M* ou *G* para 1.000, 1.000.000 e 1.000.000.000 respectivamente.

Utilizando a flag **-x** (expandir números) os números serão impressos inteiros, independente de seu tamanho.

```
iptables -L FORWARD
```

Zerando contadores. É útil zerar os contadores. Isso pode ser feito com a opção **-Z** ou **--zero**.

```
iptables -Z FORWARD
```

No exemplo acima, alguns pacotes passariam pelos comandos **-L** e **-Z**. Por isso, você pode utilizar **-L** e **-Z** em conjunto, para zerar os contadores enquanto estes são lidos.

3.6 Misturando NAT e Filtragem de Pacotes

É muito comum a utilização de `Network Address Translation` (`NAT`) em conjunto com a filtragem de pacotes. A boa notícia é que eles se misturam muito bem. Primeiramente, você projeta e constrói seu filtro de pacotes ignorando totalmente qualquer `NAT` a ser feito. As origens e destinos que o filtro de pacotes verá serão as origens e destinos reais. Por exemplo, se você fará `DNAT` para mandar conexões do endereço 1.2.3.4 porta 80 para 10.1.1.1 porta 8080, o filtro de pacotes verá pacotes indo para 10.1.1.1 porta 8080 (o destino real), e não 1.2.3.4 porta 80.

Similarmente, você pode ignorar o `masquerading`, os pacotes vão parecer que têm como origem o real endereço IP interno (por exemplo 10.1.1.1), e as respostas vão parecer que têm como destino esse mesmo endereço.

Pode-se utilizar a extensão de checagem do estado dos pacotes (`state match`) sem fazer com que o filtro de pacotes tenha qualquer trabalho extra, uma vez que `NAT` já requer acompanhamento de conexão. Para melhorar o simples exemplo de `masquerading` descrito no *NAT HOWTO* a fim de rejeitar quaisquer novas conexões vindo na interface ppp0. Fazer `NAT` `masquerade` pela interface ppp0:

```
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
# Rejeitar conexões novas (NEW) e inválidas (INVALID) de pacotes com
# destino à maquina local ou que devem ser repassados vindos de ppp0.
iptables -A INPUT -i ppp0 -m state --state NEW,INVALID -j DROP
iptables -A FORWARD -i ppp0 -m state --state NEW,INVALID -j DROP
# Habilitar IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
```

3.7 Compartilhamento de Conexões

A maneira mais fácil que existe usar o seu Linux com `gateway` é compartilhando o acesso à internet entre as suas várias sub-redes. O exemplo abaixo demonstra como fazer isso:

```
echo 1>/proc/sys/net/ipv4/ip_forward
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

No exemplo, a primeira linha diz que é ativado o encaminhamento de IP's, o que é essencial para compartilhar um link. Os três próximos comandos servem para apagar outras regras existentes. A última regra diz que é para fazer `NAT` na saída da eth0.

3.8 Fazendo NAT

O `NAT` é uma técnica usada para fazer uma rede acessar outra na qual não possui rota para ela, mas ela possua uma máquina com acesso a essa outra rede. Em outras palavras, técnica usada para uma rede local acessar a internet, contendo apenas 1 endereço IP válido. Vamos supor:

```
Rede local: 192.168.0.1 ao 192.168.0.254 - Máscara: 255.255.255.0
Endereço de Internet: 200.xxx.yyy.120 - Gateway: 200.xxx.yyy.1

Esquema do Servidor de Internet
eth0: 200.xxx.yyy.120
```

```
eth1: 192.168.0.1
Gateway: 200.xxx.yyy.1
```

Inicialmente, para o Linux habilitar a passagem de arquivos entre as interface de rede, é preciso habilitar a opção de IP-FORWARD. Para isso, digite no console:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Com isso, essa máquina já está pronta transmitir os pacotes de uma interface para outra. Agora, como a internet desconhece a existência da sua rede local, você precisa fazer com que suas estações internas acessem a internet como se fosse a máquina com acesso normal. Para isso:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -i eth1 -j MASQUERADE
```

Explicação passo-a-passo:

```
iptables = programa para controlar as requisições TCP/IP (Firewall)
-t nat = Informa que iremos colocar uma regra na Tabela de NAT (-t = tabela)
-A POSTROUTING = Adicionar uma regra no Pós-Roteamento (após passar entra as interfaces)
-s 192.168.0.0/24 = Endereço de origem (Source) das requisições
-i eth1 = Interface de Entrada (-i = entrada/IN)
-j MASQUERADE = tarefa a executar (job), que no caso é o MASCARAMENTO (MASQUERADE)
```

Para que esse esquema de NAT funcione, o IP das estações precisam fazer parta de rede local, ou seja, precisa estar entre 192.168.0.2 até o 192.168.0.254. O IP 192.168.0.1, não pode ser usado, pois já está em uso pelo servidor.

Idealmente para usar o NAT de maneira mais eficiente, juntamente com regras de roteamento (estão interconectados), devemos carregar alguns módulos como abaixo:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
/sbin/modprobe iptable_nat
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_tables
/sbin/modprobe ipt_unclean
/sbin/modprobe ipt_limit
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_REJECT
/usr/sbin/iptables -F
/usr/sbin/iptables -t nat -F
/usr/sbin/iptables -P FORWARD DROP
/usr/sbin/iptables -A INPUT -i lo -j ACCEPT
/usr/sbin/iptables -A OUTPUT -o lo -j ACCEPT
/usr/sbin/iptables -A FORWARD -s 192.168.1.0/24 -d 0/0 -j ACCEPT
/usr/sbin/iptables -A FORWARD -s 0/0 -d 192.168.1.0/24 -mstate --state ESTABLISHED,RELATED -j ACCEPT
/usr/sbin/iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -d 0/0 -j MASQUERADE
```

Nesse caso você deverá substituir o IP 192.168.1.0, pelo IP da sua rede interna.

3.8.1 Fazendo NAT 1:1

Com o NAT podemos trocar o IP não público 192.168.0.2 pelo IP público 200.222.5.2 e vice-versa. Todo pacote que vir da internet para acessar o servidor 200.222.5.2, passa pela regra PREROUTING do Firewall e logo em seguida é feito um DNAT(destination NAT) que troca o IP destino 200.222.5.2 por 192.168.0.2 e em seguida o pacote é encaminhado para o filtro, verificado nas regras de FORWARD e aí então chega ao servidor destino. Quando sair um pacote do IP 192.168.0.2 com destino a internet, o mesmo passa novamente pelas regras de FORWARD e logo em seguida passa para o POSTROUTING que faz o SNAT(source NAT) onde é trocado o IP 192.168.0.2 pelo IP 200.222.5.2 e aí sim vai para a internet. O nome dado a esse tipo de acesso é NAT 1:1.

Como puderam perceber os pacotes tanto de entrada quanto de saída não passaram pelas regras de INPUT e OUTPUT, porque essas regras só dizem respeito a pacotes direcionados ao Firewall.

Qualquer pacote cujo destino seja o Firewall, então o mesmo passa pelas regras de `INPUT`, e todo pacote que sair do Firewall passa pelas regras de `OUTPUT` do mesmo.

Para todos os pacotes de dados que passam de uma rede para outra através do Firewall, passam pelas regras de `FORWARD` e agora com o `iptables`, podemos especificar regras que controlem pacotes que entrem por uma interface em específico e saiam por outra interface.

```
iptables -A FORWARD -p tcp --dport 80 -s 0/0 -i eth0 -o eth1 -j ACCEPT
```

O **-i** indica interface de entrada (input) e o **-o** interface de saída (output). Para fazer o NAT acima bastariam 2 regras:

```
iptables -A PREROUTING  -t nat -d 200.222.5.2/32 -j DNAT --to 192.168.0.2
iptables -A POSTROUTING -t nat -s 192.168.0.2/32 -j SNAT --to 200.222.5.2
```

Não esquecer de criar o alias eth0:0 no Firewall com o IP 200.222.5.2 senão, não irá funcionar. Além disso o novo filtro traz o tão esperado STATEFUL PACKET. Com esse recurso o filtro consegue gerenciar, por exemplo, as conexões feitas pelas estações e assim não deixar aberturas para scanners atuarem como acontecia com o tão conhecido protocolo FTP e outros.

3.9 Exemplo de um firewall completo com script

```
#!/bin/bash
#### Início da função stop ####
stop()
{
 # desbilita o repasse de pacotes do kernel
 echo 0 > /proc/sys/net/ipv4/ip_forward
 # limpa regras existentes nas chains INPUT, OUTPUT E FORWARD
 /sbin/iptables -F
 # limpa as regras existentes na tabela de NAT
 /sbin/iptables -F -t nat
}
#### Fim da Função stop ####

#### Início da Função start ####
start ()
{
 # habilitando o repasse de pacotes no kernel
 echo 1 > /proc/sys/net/ipv4/ip_forward

 ######################################################################
 ########################## ROTEAMENTO ################################
 ######################################################################
 # cria rota padrão para o gateway no provedor (200.198.10.1)
 # /sbin/route add default gw 200.198.10.1 dev eth1
 ######################################################################
 ########################## REGRAS ####################################
 ######################################################################
 # Define a política padrão para cada CHAIN (DROP)
 /sbin/iptables -P FORWARD DROP
 /sbin/iptables -P INPUT DROP
 /sbin/iptables -P OUTPUT DROP
 # cria uma nova chain denominada filtro
 /sbin/iptables -N filtro
 # aceita conexões da rede interna
 /sbin/iptables -A filtro -m state --state ESTABLISHED,RELATED -j ACCEPT
 # rejeita novas conexões que não sejam da rede interna
 /sbin/iptables -A filtro -m state --state NEW ! -i eth0 -j DROP
 ######################################################################
 ################## DETECTA E REGISTRA PORTSCANNERS ###################
 ######################################################################
 # Xmas portscan
 /sbin/iptables -A filtro -p tcp --tcp-flags ALL FIN,URG,PSH -m limit --limit 2/m
--limit-burst 2 -j LOG --log-prefix "Xmas portscanner: "
 # descarta o pacote
```

```
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
      # portscanner do tipo que ativa os bits SYN e FIN
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL SYN,FIN -m limit --limit 2/m
     --limit-burst 2 -j LOG --log-prefix "SYN FIN portscanner: "
      # descarta o pacote
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL SYN,FIN -j DROP
      # portscanner do tipo que ativa os bits SYN e RST
      /sbin/iptables -A filtro -p tcp --tcp-flags SYN,RST SYN,RST -m limit --limit 2/m
     --limit-burst 2 -j LOG --log-prefix "SYN RST portscanner: "
      # descarta o pacote
      /sbin/iptables -A filtro -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
      # portscanner do tipo que ativa o bit FIN
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL FIN -m limit --limit 2/m --limit-
burst 2 -m state --state ! ESTABLISHED -j LOG –log-prefix "FIN portscanner: "
      # descarta o pacote
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL FIN -m state --state ! ESTABLISHED
-j DROP
      # portscanner do tipo que habilita todas as flags tcp
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL ALL -m limit --limit 2/m --limit-
burst 2 -j LOG --log-prefix "ALL portscanner: "
      # descarta o pacote
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL ALL -j DROP
      # portscanner do tipo que nÃ£o habilita nenhum flag
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL NONE -m limit --limit 2/m --limit-
burst 2 -j LOG --log-prefix "NONE portscanner: "
      # descarta o pacote
      /sbin/iptables -A filtro -p tcp --tcp-flags ALL NONE -j DROP

      ###################################################################
      ################## PROTEÇÃO CONTRA ATAQUES CONHECIDOS ###############
      ###################################################################
      # proteção contra synflood
      /sbin/iptables -A filtro -p tcp --syn -m limit --limit 1/s -j ACCEPT
      # proteçãoo contra ping da morte
      /sbin/iptables -A filtro -p icmp --icmp-type echo-request -m limit --limit 1/s -j
ACCEPT
      # bloqueia outras conexÃµes
      /sbin/iptables -A filtro -j DROP
      ###################################################################
      ################## POLÍTICA PARA REPASSE DE PORTAS ###################
      ###################################################################
      # libera o repasse de pacotes apenas para portas 53 udp (dns), 80 tcp (http),
      # 443 tcp (https)
      /sbin/iptables -A FORWARD -p tcp -s 192.168.1.0/24 --dport 80 -j ACCEPT
      /sbin/iptables -A FORWARD -p tcp -s 192.168.1.0/24 --dport 443 -j ACCEPT
      /sbin/iptables -A FORWARD -p udp -s 192.168.1.0/24 --dport 53 -j ACCEPT
      # aplica a chain filtro na INPUT, ou seja, os pacotes que passarem pela chain
INPUT
      # serão direcionados para a chain filtro
      /sbin/iptables -A INPUT -j filtro
      # aplica a chain filtro na FORWARD, ou seja, antes dos pacotes serem repassados
(forwarding), serãoo direcionados pela chain filtro
      /sbin/iptables -A FORWARD -j filtro
      # # # # Regras na tabela de NAT # # # #
      # mascara os pacotes que chegam na interface eth0 com destino a porta tcp 80
(http)
      /sbin/iptables -A POSTROUTING -t nat -p tcp -i eth0 --dport 80 -j MASQUERADE
      # mascara os pacotes que chegam na interface eth0 com destino a porta tcp 443
(https)
      /sbin/iptables -A POSTROUTING -t nat -p tcp -i eth0 --dport 443 -j MASQUERADE
      # mascara os pacotes que chegam na interface eth0 com destino a porta udp 53
      /sbin/iptables -A POSTROUTING -t nat -p udp -i eth0 --dport 53 -j MASQUERADE
      # disponibilizando acesso ao servidor web que roda na porta 80 e 443 do host
192.168.1.50
      /sbin/iptables -A PREROUTING -t nat -p tcp --dport 80 -j DNAT --to 192.168.1.50:80
      /sbin/iptables -A PREROUTING -t nat -p tcp --dport 443 -j DNAT --to
192.168.1.50:443
     }
```

```
#### FIM DA FUNÇÃO start ####
########################################################################
################### INICIANDO E PARANDO O SERVIDOR ################
########################################################################
if [ $# -lt 1 ]; then
 echo "$1 { start | stop | restart }";
 exit 1;
fi
if [ $1 == "start" ]; then
 echo "Iniciando o servidor firewall...";
 start;
fi
if [ $1 == "stop" ]; then
 echo "Parando o servidor firewall...";
 stop;
fi
if [ $1 == "restart" ]; then
 echo "Parando o servidor firewall..."
 stop;
 echo "Iniciando o servidor Firewall..."
start;
fi
# # # FIM DO ARQUIVO # # #
```

4 Roteamento

O roteamento pode ser feito de várias formas.

4.1 Tabelas de roteamento

No Linux pode-se ter múltiplas tabelas de roteamento no kernel. Estas tabelas são identificadas por um número variando entre 1 e 255 ou por um nome de acordo com o arquivo `/etc/iproute2/rt_tables`. Com novas tabelas pode-se criar uma flexível estrutura para implementar uma Política de Roteamento. Abaixo temos o conteúdo do arquivo `/etc/iproute2/rt_tables`:

```
k8vse:~# cat /etc/iproute2/rt_tables
#
# reserved values
#
255     local
254     main
253     default
0       unspec
#
# local
#
#1      inr.ruhep
```

- `255 local` - A tabela `local` contém rotas para endereços locais, esses endereços locais estão localizados no próprio computador, como os endereços que ele possui, a rede dele mesmo e seus `broadcast´s`. "Não manipule essa tabela ou retire-a da ordem nas regras. Ela não é útil para o roteamento de uma rede, só para o próprio computador".
- `254 main` - Tabela de roteamento principal. É nela que que ficam as rotas quando adicionamos com o comando `route` ou `ip route`. Essa tabela armazena as rotas das redes disponíveis localmente, bem como a rota padrão adicionada com o comando `route` ou `ip route`.
- `253 default` – A tabela `default` é uma tabela solitária, quase nunca é usada se você tem um `default gateway` na tabela `main`. Normalmente, quando existe um `default gateway` na tabela `main`, um pacote nunca chega a tabela `default`.

Para ver as rotas que estão estabelecidas na tabela `main`, podemos usar:

```
k8vse:~# ip route show table main
10.1.1.0/24 dev eth0  proto kernel  scope link  src 10.1.1.4
default via 10.1.1.1 dev eth0
```

Outro exemlo:

```
# ip route show table main
200.217.72.1 dev ppp0 proto kernel scope link src 200.164.113.232
10.12.0.0/24 dev eth0 proto kernel scope link src 10.12.0.1
127.0.0.0/8 dev lo scope link
default via 200.217.72.1 dev ppp0
```

Podemos ver que existem diferentes rotas dentro da tabela de roteamento `main`. Essa é a tabela principal de roteamento do `kernel` e tem como `default gateway` o ip do `roteador` que é o `200.217.72.1`, além de rotas para a rede local como a rede `10.12.0.0/24` que tem como destino de saída a interface eth0.

As regras definem em que ordem as tabelas serão consultadas. A prioridade vai de 0 até 32767 em ordem crescente, *o menor número tem maior prioridade*. As regras não tem nomes, apenas números de prioridade. Já as tabelas têm números e podem ser referenciadas por nomes no arquivo `/etc/iproute2/rt_tables`.

Para consultar as regras podemos usar:

```
# ip rule show
```

```
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default
```

Observe que os números das tabelas não têm nada a ver com número de regras. A regra 200 nada tem a ver com a tabela 200. Podemos observar também que podemos ver a prioridade das tabelas onde a tabela `main`, que possui uma prioridade maior que a tabela `default` (por ser a tabela principal de roteamento). Quanto menor for o número, maior a prioridade de roteamento.

Nesse caso a tabela `main` possui o número 32766, que é menor que 32767 e como conseqüência a tabela `main` tem prioridade maior que a tabela `default`. O valor máximo de prioridade é 32767, que por padrão já se encontra em uso, bastando apenas utilizarmos prioridades menores que 32766.

4.1.1 Adicionando Tabelas de Roteamento

Para criar uma nova tabela de roteamento, devemos inserir o arquivo `/etc/iproute2/rt_tables` e acrescentar um número e um nome. Lembre-se que por enquanto nada vai acontecer com essa tabela, pois ainda não foi inserido uma rota nela:

```
200      adsl2
```

Simplesmente isso, só acrescentar um número e um nome (nome opcional, mas facilita no hora de verificar as rotas, um nome é mais fácil de lembrar do que um número). Agora vamos ver o conteúdo dessa tabela:

```
# ip route show table adsl2
```

O resultado dessa tabela é nada, pois não definimos nada ainda. Vamos a um exemplo prático. Supondo as seguintes configurações de rede:

```
eth0       Encapsulamento do Link: Ethernet  Endereço de HW 00:11:D8:45:97:76
           inet end.: 10.1.1.4  Bcast:10.1.1.255  Masc:255.255.255.0
           endereço inet6: fe80::211:d8ff:fe45:9776/64 Escopo:Link
           UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
           RX packets:2544 errors:0 dropped:0 overruns:0 frame:0
           TX packets:3400 errors:0 dropped:0 overruns:0 carrier:0
           colisões:0 txqueuelen:1000
           RX bytes:2357404 (2.2 MiB)  TX bytes:425298 (415.3 KiB)
           IRQ:193 Memória:f7e00000-0

lo         Encapsulamento do Link: Loopback Local
           inet end.: 127.0.0.1  Masc:255.0.0.0
           endereço inet6: ::1/128 Escopo:Máquina
           UP LOOPBACKRUNNING  MTU:16436  Métrica:1
           RX packets:302 errors:0 dropped:0 overruns:0 frame:0
           TX packets:302 errors:0 dropped:0 overruns:0 carrier:0
           colisões:0 txqueuelen:0
           RX bytes:30048 (29.3 KiB)  TX bytes:30048 (29.3 KiB)

wlan0      Encapsulamento do Link: Ethernet  Endereço de HW 00:0E:2E:0F:33:97
           inet end.: 192.168.0.1  Bcast:192.168.0.255  Masc:255.255.255.0
           endereço inet6: fe80::20e:2eff:fe0f:3397/64 Escopo:Link
           UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:139 errors:0 dropped:0 overruns:0 carrier:0
           colisões:0 txqueuelen:1000
           RX bytes:0 (0.0 b)  TX bytes:15690 (15.3 KiB)
           IRQ:169 Memória:f7c00000-f7c00100
```

Na situação acima, estou com somente uma interface física eth0 com o IP 10.1.1.4 e outra wlan0 com IP 192.168.0.1 ambas classe C. Nessa situação a tabela main:

```
k8vse:~# ip route show table main
192.168.0.0/24 dev wlan0  proto kernel  scope link  src 192.168.0.1
10.1.1.0/24 dev eth0  proto kernel  scope link  src 10.1.1.4
```

```
default via 10.1.1.1 dev eth0
```

O próximo passo é ativar o roteamento no kernel:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Esse comando habilita o roteamento pela ativação de encaminhamento de IP´s (seta em 1, ou seja, ativa).

O procedimento abaixo desliga o `rp_filter`, que é um filtro contra pacotes `IP marcianos` **(martian IP)**. Esses pacotes são aqueles que deveriam ter chegado por uma interface, porém chegam por outra. O `rp_filter` visa detectar pacotes `spoofados`, o que é uma boa idéia. Porém, no nosso caso, os pacotes podem ir e voltar por mais de uma interface, a filtragem do `rp_filter` falha. Portanto, temos que desligá-lo e reforçarmos as segurança das interface privadas com regras de firewall criadas por nós. Iremos fazer esse desligamento criando um script com o seguinte conteúdo:

```
#!/bin/bash
# liberar marcianos

for i in /proc/sys/net/ipv4/conf/*/rp_filter; do
      /usr/bin/echo 0 > $i
done
```

4.1.2 Inserindo uma rota na nova tabela de roteamento

Na seção 4.1.2 nós criamos uma tabela de roteamento chamada `adsl2,` agora nós vamos inserir um rota para essa tabela, para depois então inserir uma regra nela. Para inserir uma rota `default` nessa tabela, devemos digitar:

```
ip route add default dev wlan0 via 192.168.0.1 table adsl2
```

Vejamos como ficou o conteúdo da tabela de roteamento:

```
k8vse:~# ip route show table adsl2
default via 192.168.0.1 dev wlan0
```

Pronto, a rota já foi criada nessa tabela. Falta apenas inserirmos uma regra para essa rota que acabamos de criar na tabela de roteamento adsl2.

4.1.3 Inserindo uma regra na nova tabela de roteamento

Primeiramente, vamos marcar todos os pacotes provenientes da LAN com destino aos serviços `http` com a seguinte regra de `iptables` (Capítulo 3) e logo em seguida, será inserido uma regra nessa tabela de roteamento:

```
iptables -t mangle -A PREROUTING -i eth0 -p tcp --dport 80 -j MARK --set-mark 1
```

Com isso, os pacotes já estão marcados, mas nenhum pacote vai entrar nela, pois não há regras para isso. Devemos colocar uma regra para que se um determinado pacote possuir a marca, seja verificada determinada tabela de rota:

```
ip rule add fwmark 1 lookup adsl2
```

Para que uma modificação de rota entre em funcionamento instantaneamente, é necessário digitarmos o comando abaixo para as novas regras vigorarem.

```
ip route flush cached
```

Tudo pronto, agora só falta fazer o NAT para que as máquinas da LAN possam acessar a internet, sendo que os pacotes com destino a porta 80 sairão pela interface wlan0 e os demais sairão pela interface eth0.

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
```

4.1.4 Eliminando Regras

Suponhamos que eu quero retirar a regra que incluirmos na tabela adsl2 fazendo com que todo o tráfego saia pela eth0. Observem a seguinte seqüência de comandos:

```
ip rule del fwmark 1 lookup adsl
ip route flush cached
```

Pronto. Todo o tráfego já está saindo nesse momento peleth0. Repare que só excluindo a regra que inserimos na tabela adsl2 deixa a rota `default` na tabela adsl2. Ou seja, podemos inserir quantas rotas quisermos, mas de nada adianta se não existirem regras para utilização dessas rotas.

4.1.5 Roteamento por Redes ou IP´s

Outra forma de criar uma regra, só que agora sem analise de marcação de pacote, mas sim uma informação pertinente ao pacote (IP), poderia ser feita como:

```
ip rule add from 10.12.0.20 lookup adsl2
ip rule add from 10.12.0.21 lookup adsl2
ip rule add from 10.12.0.22 lookup adsl2
ip route flush cached
```

Nesse caso, estou informando que todos os pacotes cujo IP de origem seja 10.12.0.20, 10.12.0.21 e 10.12.0.22 devem sair pelo segundo link de internet wlan0.

4.2 Esquema de Roteamento com Firewall e NAT

4.3 Exemplo Pratico Completo

```
#=====================================================================
#                  ESQUEMA DE CONFIGURAÇÃO DE ROTEAMENTO
#
# Autor___: Alexandre Stürmer Wolf
# Data____: 30/06/2006
# Função__: Permitir que as redes 192.168.1.0 e 192.168.52.0 possam
#           acessar a internet usando proxy transparente pelo  SQUID
#           utilizando um link PPP para acessos em  geral e um  link
#           ADSL para acessos de programas P2P.
#
#
#               210.44.123.45                  230.34.45.x
#                 +------+                       +-----+
#                 | ADSL |                       | PPP |
#                 +------+                       +-----+
#       10.1.1.1     |               230.34.45.1   |
#                    |                             |
#                    \                             /
#                     \                           /
#                      \                         /
#                       \                       /
#                        \                     /
#      eth1  10.1.1.3     +--------------------+ 230.34.45.9  eth0
#                         |   ROTEADOR LINUX   |
#                         |      DEBIAN        |
#                         +--------------------+
#                        /                      \
#                       /                        \
#                      /                          \
#        eth2  192.168.1.1               eth3   192.168.52.1
#
#
#=====================================================================

#=====================================================================
# RESSETANDO AS CONFIGURACOES E REINICIANDO COM ROTEAMENTO PADRAO
```

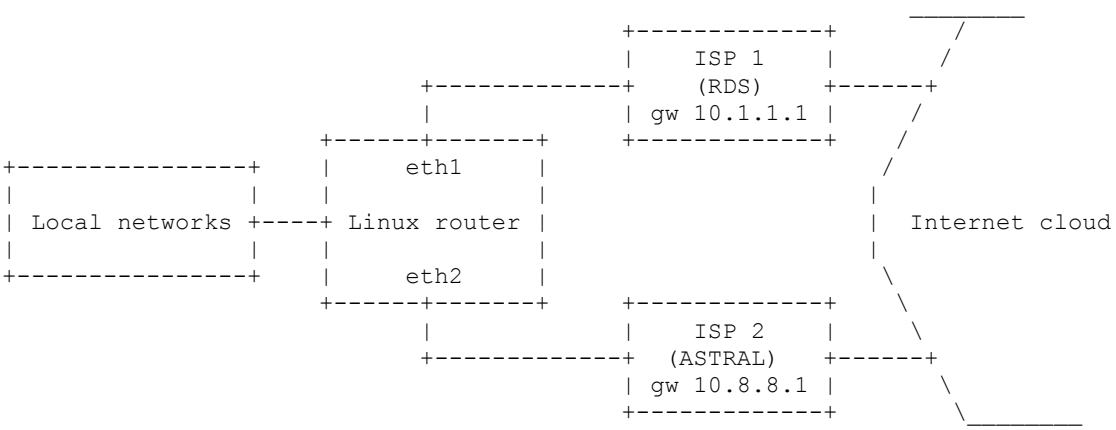
```
#====================================================================
echo LIMPANDO REGRAS ANTERIORES

iptables -F
iptables -X
iptables -Z
iptables -t nat -F

echo LEVANTANDO AS ETHS AUXILIARES PARA NAT-ONE-NAT

ifconfig eth0:0 230.34.45.55 netmask 255.255.255.0

echo INICIANDO O ROTEAMENTO BASICO

iptables -P INPUT ACCEPT
iptables -F INPUT
iptables -P OUTPUT ACCEPT
iptables -F OUTPUT
iptables -P FORWARD DROP
iptables -F FORWARD

modprobe ipt_MASQUERADE
modprobe iptable_nat
modprobe ip_conntrack
modprobe ip_conntrack_ftp
modprobe ip_tables
modprobe iptable_filter
modprobe ipt_limit
modprobe ipt_LOG
modprobe ipt_REJECT
modprobe ip_nat_ftp


echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

#====================================================================
# FAZ NAT TRANSPARENTE PARA O SQUID --PROXY--
# DIRECIONANDO A PORTA 80 PARA 3128
# Se não for a %$# social
#====================================================================
echo REDIRECIONANDO A PORTA 80 PARA O SQUID PARA PROXY TRANSPARENTE

iptables -t nat -A PREROUTING -i eth2 -d ! 200.201.174.0/24 -p tcp --dport 80 -j
REDIRECT --to-port 3128
iptables -t nat -A PREROUTING -i eth3 -d ! 200.201.174.0/24 -p tcp --dport 80 -j
REDIRECT --to-port 3128

iptables -A FORWARD -p tcp -i eth0 --dport 80 -d 200.201.174.0/24 -j ACCEPT
iptables -A FORWARD -p tcp -i eth0 --dport 443 -d 0/0 -j ACCEPT

#iptables -t nat -A PREROUTING -i eth2 -p tcp --dport 80 -j REDIRECT --to-port 3128
#iptables -t nat -A PREROUTING -i eth3 -p tcp --dport 80 -j REDIRECT --to-port 3128

#====================================================================
# NAT DE PORTAS PARA CLIENTES ESPECIAIS
# DIRECIONANDO PORTAS
#====================================================================
echo REDIRECIONANDO PORTAS PARA IP DE CLIENTES

# Cliente 192.168.1.46
iptables -A INPUT -p tcp -i eth0 --destination-port 4899 -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --destination-port 3390 -j ACCEPT

iptables -A PREROUTING -t nat -p tcp -d 230.34.45.9 --dport 4899 -j DNAT --to
192.168.1.46:4899
iptables -A PREROUTING -t nat -p tcp -d 230.34.45.9 --dport 3390 -j DNAT --to
192.168.1.46:3390
```

```
iptables -A FORWARD -p tcp -i eth0 --dport 4899 -d 192.168.1.46 -j ACCEPT
iptables -A FORWARD -p tcp -i eth0 --dport 3390 -d 192.168.1.46 -j ACCEPT

#=====================================================================
# ATIVANDO ESQUEMA DE NAT 1 : 1
#=====================================================================
echo ATIVANDO REGRAS DE NAT 1 p/ 1

iptables -t nat -A PREROUTING -d 230.34.45.55 -i eth0 -j DNAT --to-destination
192.168.52.55
iptables -t nat -A POSTROUTING -s 192.168.52.55 -o eth3 -j SNAT --to-source
230.34.45.55

#=====================================================================
# FAZ NAT DE TUDO QUE ENTRA E QUER SAIR APONTANDO PARA PPP E ADSL
# ONDE ETHO E O PPP E ETH1 E O ADSL, COM mtu ENTRE 1400 E 1536
#=====================================================================
echo HABILITANDO NAT INCONDICIONAL PARA PPP E ADSL

iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 1400:1536 -j
TCPMSS --clamp-mss-to-pmtu

iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

iptables -A FORWARD -s 192.168.1.0/24 -d 0/0 -j ACCEPT
iptables -A FORWARD -s 192.168.52.0/24 -d 0/0 -j ACCEPT
iptables -A FORWARD -s 10.1.1.0/24 -d 0/0 -j ACCEPT


iptables -A FORWARD -s 0/0 -d 192.168.1.0/24 -mstate --state ESTABLISHED,RELATED -j
ACCEPT
iptables -A FORWARD -s 0/0 -d 192.168.52.0/24 -mstate --state ESTABLISHED,RELATED
-j ACCEPT
iptables -A FORWARD -s 0/0 -d 10.1.1.0/24 -mstate --state ESTABLISHED,RELATED -j
ACCEPT

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

#=====================================================================
# LIBERANDO PORTAS ACIMA DE 1024 INCONDICIONALMENTE
#=====================================================================
echo LIBERANDO PORTAS ACIMA de 1024

iptables -A INPUT -p tcp -m tcp --sport 1024:65535 --dport 1024:65535 -m state --
state NEW -j ACCEPT
iptables -A INPUT -p udp -m udp --sport 1024:65535 --dport 1024:65535 -m state --
state NEW -j ACCEPT

iptables -A INPUT -p tcp --sport 1024:65535 --dport 1024:65535 -m state --state
ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 1024:65535  -m state --state
ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -p tcp --sport 1024:65535 -j ACCEPT
iptables -A INPUT -p udp --sport 1024:65535 -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 -j ACCEPT
iptables -A INPUT -p udp --dport 1024:65535 -j ACCEPT

#=====================================================================
# PERMITE QUE SE MANTENHAM OU SE CONECTEM REQUISICOES FEITAS POR
# DENTRO DA REDE TANTO DO ADSL QUANTO PPP PARA AS REDES INTERNAS
#=====================================================================
echo RECEBIMENTO DE DADOS PARA DENTRO

iptables -A FORWARD -i eth0 -o eth3 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i eth0 -o eth2 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
#====================================================================
# MARCA OS PACOTES COM DESTINO A DETERMINADAS PORTAS PARA POSTERIOR
# ENCAMINHAMENTO AO ADSL PELO BLOQUEIO DE SAIDA DOS DADOS PELO PPP
#====================================================================
echo MARCANDO PACOTES COM DESTINO AO ADSL

# E-MULE, E-DONKEY e CLONES
iptables -A PREROUTING -t mangle -p tcp --dport 4661:4676 -j MARK --set-mark 3
iptables -A PREROUTING -t mangle -p udp --dport 4661:4676 -j MARK --set-mark 3

iptables -A PREROUTING -t mangle -p tcp --dport 5221 -j MARK --set-mark 3
iptables -A PREROUTING -t mangle -p udp --dport 5221 -j MARK --set-mark 3

iptables -A PREROUTING -t mangle -p tcp --dport 5661 -j MARK --set-mark 3
iptables -A PREROUTING -t mangle -p udp --dport 5661 -j MARK --set-mark 3

#====================================================================
# DESLIGANDO O TRATADOR rp-filter PARA NAO DESCARTAR O ENCAMINHAMENTO
#====================================================================
echo MARCANDO PACOTES COM CONTROLE DO ADSL

for eee in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 0 > $eee
 done

#====================================================================
# ATIVANDO AS REGRAS PARA O ENCAMINHAMENTO DE ACORDO COM A MARCACAO DOS
# PACOTES ANTERIORES, ANALISANDO PELA MARCACAO COM O NUMERO 3
#====================================================================
echo ATIVANDO AS REGRAS PARA OS PACOTES

eth1 ip rule add fwmark 3 lookup 30 prio 30

eth1 ip route add default via 10.1.1.1 table 30

#====================================================================
# INFORMANDO A ROTA PADRAO (CASO NAO CAIA NAS REGRAS)
#====================================================================
echo ATIVANDO A ROTA PADRAO

ip route add default via 230.34.45.1 proto static

#====================================================================
# ATIVANDO AS NOVAS REGRAS
#====================================================================
echo LIMPANDO A MEMORIA COM AS ROTAS PENDENTES

ip route flush cache

#====================================================================
# ATIVANDO MAIS REGRAS PARA PROTECAO
#====================================================================
echo ATIVANDO REGRAS DE PROTECAO Baterias de Defesa

# Syn-flood
iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT

# Port Scanner
iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j
ACCEPT

# Ping da Morte
iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT


# Marcacao para o CBQ
```

```
#iptables -t mangle -I FORWARD -s 192.168.1.1/32 -j MARK --set-mark 1
#iptables -t mangle -A PREROUTING -d 192.168.1.1/32 -j MARK --set-mark 1
```

4.4 Roteamento por Redes



Execute:

```
echo 1 RDS >> /etc/iproute2/rt_tables
echo 2 ASTRAL >> /etc/iproute2/rt_tables
```

O conteúdo do arquivo /etc/iproute2/rt_table depois dos Comando s:

```
#
# reserved values
#
255     local
254     main
253     default
0       unspec
#
# local
#
#1      inr.ruhep
1 RDS
2 ASTRAL
```

Nesse momento temos tres tabelas de rotas, RDS, ASTRAL e a rota principal. O próximo passo é criar rotas para a tabela RDS.

```
ip route add default via 10.1.1.1 dev eth1 table RDS
ip rule add from 10.11.11.0/24 table RDS
ip rule add from 10.12.12.0/24 table RDS
```

Agora para a tabela ASTRAL:

```
ip route add default via 10.8.8.1 dev eth2 table ASTRAL
ip rule add from 10.22.22.0/24 table ASTRAL
ip rule add from 10.33.33.0/24 table ASTRAL
```

Vejamos as rotas:

```
ip route show table ASTRAL
ip route show table RDS
ip route show table main
```

Para ver a rota principal, você pode usar route -n, agora vejamos as tabelas de rotas:

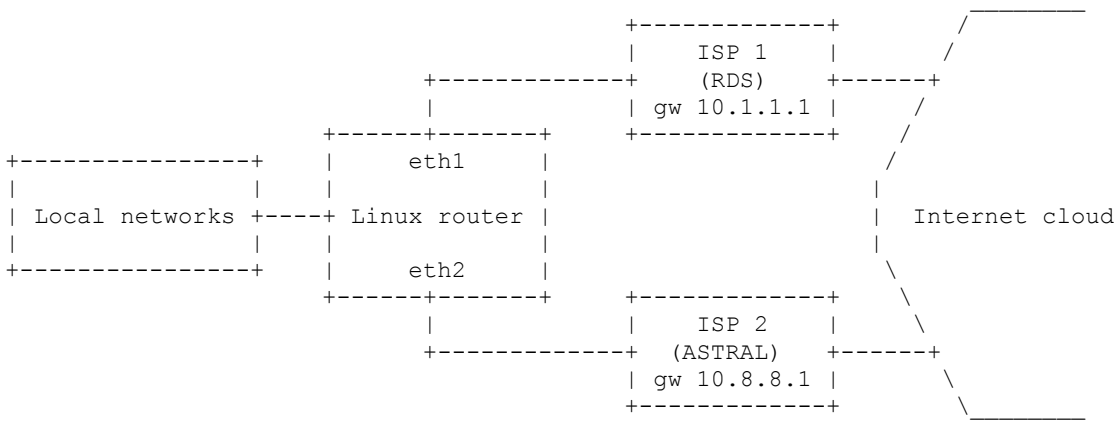
```
ip rule show                    # todas
ip rule show | grep ASTRAL      # somente ASTRAL
```

```
ip rule show | grep RDS         # somente RDS
```

Os pacotes que vem de 10.11.11.0/24 e 10.12.12.0/24 vão até a tabela de roteamento RDS. Similarmente os pacotes que chegarem das redes 10.22.22.0/25 e 10.33.33.0/24 vão para o gateway ASTRAL. Os pacotes que não se enquadram nas regras passam pelo rota principal.

4.5 Roteamento por Portas

A rota dos pacotes que tem como destino a porta 22/tcp e 80/tcp devem ir para o gateway ASTRAL.



Dado o conteúdo da tabela `/etc/iproute2/rt_table`:

```
#
# reserved values
#
255     local
254     main
253     default
0       unspec
#
# local
#
#1      inr.ruhep
1 RDS
2 ASTRAL
```

Nesse caso, devemos marcar os pacotes que possuem 22 e 80 como a porta destino, nesse caso devemos utilizar a tabela MANGLE.

```
iptables -A PREROUTING -t mangle -i eth0 -p tcp --dport 22 -j MARK --set-mark 1
iptables -A PREROUTING -t mangle -i eth0 -p tcp --dprot 80 -j MARK --set-mark 2
```

Para a tabela RDS:

```
ip route add default via 10.1.1.1 dev eth1 table RDS
```

Para tabela ASTRAL:

```
ip route add default via 10.8.8.1 dev eth2 table ASTRAL
```

O próximo passo é fazer o roteamente baseado nas marcações dos pacotes. Para RDS temos:

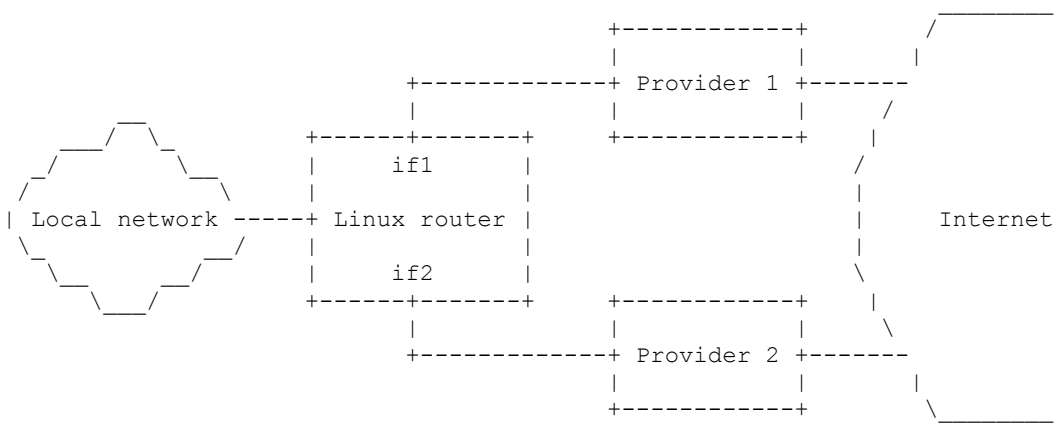
```
ip rule add from all fwmark 1 table RDS
```

Para ASTRAL temos:

```
ip rule add from all fwmark 2 table ASTRAL
```

4.6 Roteamento com multi-uplinks

A configuração mais comum é a seguinte, onde temos dois provedores conectados em uma máquina que fornece o link para um rede local.



4.6.1 Repartir a conexão

Para facilitar, vamos usar nomes simbólicos, $IF1 será o nome da primeira interface (if1 no desenho) e $IF2 o nome da segunda interface. $IP1 é o endereço associado a $IF1 e $IP2 é o endereço associado a $IF2. Nesse caso $P1 será o endereço IP do gateway do provedor 1, e $P2 o endereço IP do gateway do provedor 2. Finalmente, o $P1_NET será o IP de entrada da rede $P1 e $P2 o endereço IP de entrada para a rede SP2.

Primeiramente vamos criar 2 tabelas de rotas chamadas de T1 e T2. Elas serão adicionadas em `/etc/iproute2/rt_tables`.

```
ip route add $P1_NET dev $IF1 src $IP1 table T1
ip route add default via $P1 table T1
ip route add $P2_NET dev $IF2 src $IP2 table T2
ip route add default via $P2 table T2

ip route add $P1_NET dev $IF1 src $IP1
ip route add $P2_NET dev $IF2 src $IP2
```

Por último a rota padrão:

```
ip route add default via $P1
```

Próximo passo é ativar as regras de roteamento.

```
ip rule add from $IP1 table T1
ip rule add from $IP2 table T2
```

4.6.2 Load balancing

O Load Balance é uma maneira bastante eficiente de se utilizar 2 links. Aproveitando a idéia do exemplo anterior:

```
ip route add default scope global nexthop via $P1 dev $IF1 weight 1 \
nexthop via $P2 dev $IF2 weight 1
```

Nesse caso a os dois links possuem o mesmo peso, assim ambos possuem a mesma funcionalidade. Podemos ainda informar pesos diferentes, onde um link é preferencial e o outro auxiliar (opção weight=peso)

Observe que a \ significa que o comando continua na próxima linha.

5 Gerando Análise e Estatísticas

5.1 Webalizer (analise web)

O Webalizer é um gerador de estatísticas de acesso para o servidor web. O Apache, por si só, loga todos os acessos feitos ao servidor, incluindo as páginas acessadas, o tráfego gerado, os navegadores e sistemas operacionais usados pelos clientes, entre outras informações úteis para entender os hábitos e interesses de seus visitantes.

Com o Apache funcionando, é simples instalar o Webalizer:

```
apt-get install webalizer
```

Ao contrário do Apache, o Webalizer não é um serviço que fica residente, mas sim um executável que precisa ser chamado cada vez que quiser ver a página de estatísticas atualizada (somente como root).

```
webalizer
```

Por padrão, a página de estatísticas é armazenada na pasta webalizer/, dentro do seu servidor web. Se o Apache estiver configurado para armazenar as páginas dentro do diretório /var/www, então as estatísticas vão para a pasta local /var/www/webalizer (depende da versão do Apache (seção 2.1)).

O arquivo de configuração do Webalizer é o /etc/webalizer.conf. É importante que você revise o arquivo de configuração, indicando pelo menos a localização correta do arquivo de log do Apache e altere a pasta onde as estatísticas ficarão armazenadas, caso não queira que elas fiquem disponíveis ao público. Você pode armazená-las numa pasta isolada no servidor web, como por exemplo /var/webalizer, de forma que elas fiquem disponíveis apenas localmente, ou através de um script. As duas opções dentro do arquivo são:

```
LogFile /var/log/apache/access.log
OutputDir /var/www/webalizer
```

5.2 Sarg (analise Squid)

O Sarg é um interpretador de logs para o Squid (seção 2.9), assim como o Webalizer para o Apache. Sempre que executado ele cria um conjunto de páginas, divididas por dia, com uma lista de todas as máquinas que foram acessadas e a partir de cada máquina da rede veio cada acesso.

Caso você tenha configurado o Squid para exigir autenticação, ele organiza os acessos com base nos logins dos usuários, caso contrário ele mostra os endereços IP das máquinas.

Por padrão ele gera um conjunto de páginas html dentro da pasta /var/www/squid-reports/ que você pode visualizar através de qualquer navegador. Os acessos são organizados por usuário (caso esteja usando autenticação) ou por IP, mostrando as páginas acessadas por cada um, quantidade de dados transmitidos, tempo gasto em cada acesso, tentativas de acesso bloqueadas pelos filtros de conteúdo e outras informações. Para visualizar esse relatório, simplesmente chame o seu *web-browser* e digite o seguinte endereço var/www/squid-reports/. Para instalar o Sarg:

```
apt-get install sarg
```

O Sarg não é um daemon que fica residente (digitar sarg), você precisa apenas chama-lo quando quiser atualizar os relatório, se você quiser automatizar esta tarefa, pode usar o cron para que ele seja executado automaticamente todos os dias ou uma vez por hora por exemplo.

Você pode alterar a pasta onde são salvos os relatórios, limitar o acesso às estatísticas e alterar várias opções visuais no arquivo de configuração do Sarg, que é o /etc/squid/sarg.conf.

5.3 NTOP

O Ntop permite o analise do fluxo de dados pela rede. Para instalar basta digitar no console:

```
apt-get install ntop
```

Será feito o download e depois pedirá para confirmar algumas coisas básicas, pronto, ele provavelmente abriu uma porta 3000 para acesso local, bastando agora digitar no browser a seguinte URL: `http://localhost:3000`.

Eventualmente, é necessário dar direitos de escrita para o usuário nobody no diretório `/var/lib/ntop`, para descobrir quais arquivos deverão ter acesso, digite ntop no seu prompt, que será informado arquivo por arquivo.

6 Segurança

Pelo fato do Linux ser um sistema com capacidades de administração remota, aliado às "receitas de bolo" encontradas na internet e a configuração padrão da maioria dos servidores ser funcionais, deixam muitas brechas para usuários inescrupulosos invadir sua máquina ou utilizá-la sem seu consentimento.

Se isso já não bastasse, existe ainda artigos técnicos que ensinam como atacar servidores com a finalidade de ensinar a segurança. O estranho disso é que as pessoas somente se importam com o ataque e não a defesa. Não existe um servidor 100% seguro, mas podemos utilizar várias ferramentas para tornar ele mais seguro e não tão limitado.

A segurança não é uma mas sim três vertentes, lógica, operacional e física. Não adianta ter as melhores e mais avançadas técnicas de segurança lógica contra ataques, se um funcionário pode ter acesso ao servidor. Segundo estatísticas as maiores falhas de segurança são causadas pelo próprio pessoal da empresa (operacional). Imagine, que sua empresa possua uma rede onde todas pessoas são realmente confiáveis, mas no cabeamento da rede de alguma forma, pessoas inescrupulosas possuem acesso, assim temos problemas de ordem física. Portanto, antes de criar centenas de check-lists, leve em consideração as três bases da segurança.

No capítulo 5, são apresentadas algumas ferramentas de análise de conexão, você pode utilizar-las para verificar a condição de segurança de suas rede, no entanto muito cuidado, assim como essas ferramentas podem ser usadas por você para encontrar brechas, caso você as deixe habilitadas para a internet ou para outros usuários, as mesmas podem ser a maior brecha de todas, portanto, muita atenção.

6.1 Consideração sobre a segurança lógica

A segurança começa pela instalação do seu sistema. Se você tem uma máquina sem configurações especiais de segurança e quer torná-la segura, uma opção interessante é reinstalá-la completamente. Mas antes de instalar use somente mídias de repositórios confiáveis. Quando instalar um servidor use somente a versão `stable`, do repositório oficial da `Debian`. Ao instalar o sistema considere:

- Só exponha o sistema à Internet após completar todos os procedimentos de segurança. Há muitos relatos de máquinas invadidas durante o processo de configuração inicia.
- Se a sua distribuição permitir, opte por uma instalação personalizada, e escolha apenas os pacotes que você sabe que irá usar.
- Monte seu esquema de partições adequadamente. Se você for rodar algum serviço que gere armazenamento de dados em disco (uma proxy web, um servidor de mail, um servidor de news), além de configurar o consumo de memória e disco, de preferência à criar uma partição separada, evitando assim que os arquivos dos seus processos servidores possam lotar o espaço de armazenamento da máquina, tirando-a de operação.
- Após instalar todos os pacotes, **instale todas as atualizações de segurança** disponíveis no web site do seu fornecedor.

6.2 Serviços desnecessários

Uma instalação padronizada de sistema operacional costuma habilitar uma série de serviços dos quais você não precisa, e que podem vir a servir como ponto de acesso para um invasor.

Analise todos scripts que fazem parte da inicialização e verifique se realmente são necessários, se você não sabe para que serve, procure na internet. Lembre-se que existem scripts responsáveis por ativar diversos serviços, como por exemplo, o `/etc/inetd.conf`. Não é necessário apagar a linhas do arquivo de configuração, basta colocar um # antes do comando, que isso significa comentário. Ferramentas como o ksysv (caso não esteja instalado, use `apt-get install ksysv`) ou o "*gksu services-admin*" podem ajudar a verificar quais programas são disparados, usando um ferramenta gráfica.

De modo geral, os init scripts definem quais serviços serão inicializados no momento do boot. Alguns serviços são aparentemente inócuos do ponto de vista de uma possível invasão (sound, random, apmd), enquanto outros claramente oferecem algum raio de ação para o potencial invasor (portmap, xntpd, netfs, rstatd, rusersd, rwalld, rwhod, bootparamd, squid, yppasswd, ypserv, dhcpd, snmpd, named, routed, lpd, gated, httpd, xfs, linuxconf e muitos outros).

6.3 Conselhos Genéricos

Procure na internet dicas de como melhorar a segurança do seu servidor, mas procure em locais confiáveis. Dentre algumas dicas:

- Habilite logs para poder analisar o funcionamento do servidor.
- Utilize uma estratégia de backup confiável.
- Instale ferramentas de análise e rotação de logs.
- Instale ferramentas de garantia de integridade.
- Certifique-se de estar usando shadow passwords (não usar senhas criptografadas visíveis no `/etc/passwd`, atualmente este recurso já é padrão).
- Se você estiver com o servidor de ftp habilitado, configure-o adequadamente, não permita o acesso de usuário root, restrinja a árvore de diretórios, reveja os direitos de acesso anônimos, considere a possibilidade das senhas de acesso estarem sendo monitoradas em sua rede local.
- Desabilite o telnet**,** use o ssh, de preferência se logue como usuário comum e após conectado, use o comando `su` para se tornar root.
- Habilite os tcp wrappers (tcpd) e configure adequadamente os arquivos de restrição e permissão de acesso (`/etc/hosts.deny` e `/etc/hosts.allow`)
- Habilite filtros de pacotes usando um firewall de sua preferência, não permita que outras máquinas iniciem conexões com a sua, exceto para as portas que você explicitamente definiu.
- Bloqueie o acesso às portas dos seus serviços locais (xfs, X, servidor web utilizado para testes, entre outras). A documentação de qualquer filtro de pacotes explica claramente como fazer isto.

Referência Bibliográficas

É importante salientar que os conteúdos aqui presentes, são uma compilação adaptada de vários autores, que provavelmente utilizaram materiais de outros autores e assim conseqüentemente. Para não ser injusto, prefiro informar os principais sites:

www.debian.org
http://www.guiadohardware.net // muitos materiais da autoria de Carlos E. Morimoto
http://www.linuxit.com.br
http://www.netfilter.org/
http://focalinux.cipsga.org.br/
http://us1.samba.org/samba/
http://www.squid-cache.org/
http://www.openit.com.br/

http://www.google.com/linux?q=&restrict=linux
www.google.com.br

Entre muitos outros, principalmente em faq´s e sites de discussão.

Hoje a internet é um grande livro aberto com muito lixo, você deve filtrar muito bem as informações antes de colocar em prática. Não acredite em tudo que ler, valide as informações, muitos dos sites acima, são confiáveis, mas sujeitos a erros, inclusive esse tutorial, portanto, sempre experimente antes de aplicar, e divulgue o seu conhecimento.

Obrigado, Alexandre Stürmer Wolf (as_wolf@terra.com.br)