

Criando um Servidor

Básico no Linux

Configurando um servidor LAMP	3
Instalando um servidor LAMP	3
Instalando o Apache	3
Configuração básica	4
Virtual Hosts	5
Gerando estatísticas	10
Instalando o suporte a PHP	11
Instalando o MySQL	13
Instalando o phpBB	17
Configurando um servidor de e-mails	22
Instalando o Postfix	23
Cadastrando usuários e Configurando	29
Cadastrando usuários	29
Configurando	29
Instalando um webmail	32
Autenticando os clientes e Ativando o TLS	37
Autenticando os clientes	37
Ativando o TLS	40
Adicionando um antivírus	43
Configurando o DNS Reverso	47
Configurando um servidor Proxy com o Squid	51
Instalando o Squid	56
Criando uma configuração básica	58
Configurando o cache de páginas e arquivos	60
Adicionando restrições de acesso	63
Proxy com autenticação	67
Configurando um proxy transparente	69
Usando o Sarg para monitorar o acesso	71
Samba, parte 1: Instalação e configuração usando o swat	73
Instalando	77
Cadastrando os usuários	80
Usando o Swat	82
Permitindo que os usuários compartilhem pastas	94
Samba, parte 2: Configuração avançada do Samba	96
Ajustando as permissões de acesso	100
A seção [global]	109
A seção [homes]	115
A conta guest	117
Auditando os acessos	119
Backends: smbpasswd ou tdbsam	121
Portas e firewall	123
Samba, parte 3: Usando o Samba como PDC	124
Logando Clientes Windows	131
Cadastrando as máquinas sem usar a conta de root	136
Ajustando as permissões locais	136
Logando Clientes Linux	139
Usando o PDC para autenticação local	142
Samba, parte 4: Compartilhando impressoras no Samba	147
Disponibilizando drivers de impressão para os clientes	156
Compartilhando através do Cups	167
Como criar um firewall e compartilhar conexão usando IPTables	171

Configurando um servidor LAMP

Junto com os servidores de e-mail, os servidores web e FTP são provavelmente os mais comuns. A eles podemos adicionar os servidores DNS, que convertem os nomes de domínio em endereços IP. Ao colocar no ar um servidor com domínio registrado, juntamente com os demais serviços, você precisa configurar o servidor DNS para responder pelo seu domínio.

Nos primórdios da internet usávamos páginas html estáticas e scripts CGI. O Apache em si continua oferecendo suporte apenas a estes recursos básicos, mas ele pode ser expandido através de módulos, passando a suportar scripts em PHP, acessar bancos de dados MySQL entre inúmeros outros recursos.

O Apache continua fornecendo apenas páginas html estáticas. Sempre que for solicitada uma página em PHP ou outra linguagem, entra em ação o módulo apropriado, que faz o processamento necessário e devolve ao Apache a página html que será exibida. Entram em ação então os gestores de conteúdo e fóruns, que combinam os recursos do PHP com um banco de dados como o MySQL, acessado através dele. A combinação de tudo isso forma a solução que é popularmente chamada de "**LAMP**" (**L**inux **A**pache **M**ySQL **P**HP).

Instalando um servidor LAMP

Atualmente quase 70% dos servidores web do mundo rodam o **Apache**, a maior parte deles sobre o Linux. O Apache é um dos servidores web mais antigos, seguro e com inúmeros módulos que adicionam suporte aos mais exóticos recursos.

Ao longo de sua história, o Apache vem sucessivamente derrotando todos os servidores web proprietários. O próprio IIS da Microsoft, que alguns anos atrás parecia um concorrente fortíssimo, hoje em dia é usado em pouco mais de 10% dos servidores.

A maioria das páginas atuais utiliza uma estrutura em **PHP**, freqüentemente com um banco de dados **MySQL** ou **PostgreSQL**. Existem inclusive muitos sistemas prontos, como o **phpBB** (fórum) e o **PHP Nuke** (e derivados) para gerenciamento de conteúdo, que podem ser instalados sem muita dificuldade depois que o servidor web já estiver rodando.

Outro recurso muito usado é a encriptação de páginas em SSL, necessário para a criação de páginas seguras (usadas em lojas virtuais, por exemplo) e um sistema de estatísticas de acesso como o Webalizer.

Por padrão, o Apache utiliza a pasta `"/var/www"` para armazenar os arquivos do site, mas isto pode ser mudado no arquivo de configuração, que vai na pasta `"/etc/apache"`.

Instalando o Apache

A primeira escolha é entre instalar o Apache 2, ou o Apache 1.3, que ainda é usado por muita gente. O Apache 2 traz muitas vantagens, sobretudo do ponto de vista do desempenho, mas por outro lado ele é incompatível com os módulos compilados para o

Apache 1.3 e muitas opções de configuração são diferentes. A questão dos módulos não chega a ser um grande problema hoje em dia, pois todos os principais módulos já foram portados, mas muita gente que aprendeu a configurar o Apache 1.3 se sente mais confortável com ele e por isso continua usando-o até hoje, apesar das vantagens da nova versão.

Muitas distribuições continuam oferecendo as duas versões, de forma a satisfazer os dois públicos. No Debian, por exemplo, o Apache 1.3 é instalado através do pacote "**apache**", enquanto o Apache 2 é instalado através do "**apache2**".

Junto com o Apache propriamente dito, é interessante instalar também o pacote "**apache-ssl**", que adiciona suporte a páginas seguras (https).

```
# apt-get install apache
# apt-get install apache-ssl
```

Ao instalar o Apache 2, o suporte a SSL é instalado automaticamente junto com o pacote principal:

```
# apt-get install apache2
```

Depois de instalar os pacotes, inicie o serviço com o comando `"/etc/init.d/apache start"`.

Ao utilizar o Apache 2 o comando fica: `"/etc/init.d/apache2 restart"`

Em muitas distribuições o serviço do Apache chama-se "**httpd**" e não "apache" como no Debian. Este é um pedido da própria Apache Foundation, que além do servidor web desenvolve outros projetos, também distribuídos sob a marca "Apache". Neste caso o comando fica `"/etc/init.d/httpd 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://seu.servidor/index.html**" é na verdade o arquivo `"/var/www/index.html"`.

Configuração básica

A maior parte da configuração do Apache 1.3 pode ser feita através de um único arquivo, o **httpd.conf**, que no Debian pode ser encontrado no diretório `/etc/apache/`. Ao utilizar o Apache 2, o arquivo passa a ser o `"/etc/apache2/apache2.conf"`.

Observe que, assim como todos os arquivos de configuração, você precisa editá-lo como root. Para isso abra um terminal e rode o comando "su", forneça a senha de root e depois abra o arquivo com o comando:

```
# kedit /etc/apache/httpd.conf
```

A primeira configuração importante é a (ou as) portas TCP que serão usadas pelo servidor. Por default, a porta é a **80**, mas alguns serviços de banda larga, como por exemplo o Speedy da Telefonica bloqueiam esta porta, obrigando os usuários a manter seus servidores em portas alternativas. Você pode também alterar a porta para manter o seu servidor um pouco mais secreto, principalmente se for utilizada uma porta acima de 1024, já que, além do endereço IP ou domínio, os visitantes precisariam saber também a porta do servidor.

A configuração da porta está perto do final do arquivo, na linha:

```
Port 80
```

(use o localizar do editor de textos para encontrá-la mais facilmente).

Veja que por default o Apache escuta a porta 80. 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/apache restart"` ou `"service httpd restart"`.

Lembre-se de que ao alterar a porta os visitantes precisarão incluir o novo número no endereço. Se você for utilizar a porta 1080, por exemplo, todos deverão acessar o endereço `"http://seu.dominio.com:1080"`.

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" que configuramos acima. Para que ele escute também nas portas 1080 e 2480, 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"`.

Virtual Hosts

Outro recurso suportado pelo Apache e muito usado, é a possibilidade de hospedar vários sites no mesmo servidor (**shared hosting**). Mais de 70% dos sites da internet são hospedados desta forma econômica.

Neste caso, os arquivos de cada site ficam guardados numa pasta diferente e o servidor se encarrega de direcionar cada visitante ao site correto. Servidores como os dos serviços de hospedagem gratuita chegam a hospedar mais de 10.000 sites num único servidor Apache usando este recurso.

Existem duas formas de fazer isso. A primeira é ter um servidor com vários endereços IP e vincular cada site a um endereço (**IP-Based**). Este sistema é pouco usado, pois atualmente os endereços IP válidos são um recurso escasso e valioso. A segunda forma (de longe a mais usada) é ter um único endereço IP válido e vincular cada site a um nome de domínio (**Name-Based**).

Vamos ver primeiro a opção com múltiplos endereços IP que é a mais simples, e em seguida a com vários nomes:

IP-Based: Esta opção é útil caso você tenha mais de um link no mesmo servidor. Você pode usar um único servidor para duas linhas ADSL, ou duas linhas T1, por exemplo, ou pode ainda ter uma única placa de rede configurada para receber conexões em vários endereços IP, usando aliases.

Para criar aliases para sua placa de rede, basta usar o **ifconfig**, informando a placa de rede que receberá o alias (eth0, eth1, etc.) e o endereço IP em que ela passará a escutar. O alias é apenas um apelido; ele não altera a configuração original da placa de rede, apenas faz com que ela passe a se comportar como se fosse várias placas, escutando em vários endereços diferentes. É sem dúvida um recurso muito interessante.

Se você deseja que a sua interface eth0 passe a escutar também nos endereços 220.177.156.2, 220.177.156.3 e 220.177.156.4, os comandos seriam:

```
# ifconfig eth0:0 220.177.156.2
# ifconfig eth0:1 220.177.156.3
# ifconfig eth0:2 220.177.156.4
```

Um detalhe importante é que os aliases são desativados sempre que o servidor é reiniciado. Para que a alteração seja permanente, é necessário adicionar os comandos no arquivo `"/etc/init.d/bootmisc.sh"` ou `"/etc/rc.d/rc.local"` para que eles sejam executados a cada boot.

No Apache, basta criar seções no arquivo `"httpd.conf"`, indicando as configurações de cada site, como por exemplo:

```
ServerAdmin roberto@usuario.com
DocumentRoot /var/www/roberto/www
ServerName www.roberto.com.br
ErrorLog /sites/roberto/logs/error_log
TransferLog /sites/roberto/logs/access_log
```

```
ServerAdmin maria@usuario.com
DocumentRoot /var/www/maria/www
ServerName www.maria.com.br
```

ErrorLog /sites/maria/logs/error_log
TransferLog /sites/maria/logs/access_log

Criamos aqui a configuração para dois sites distintos, um no endereço 220.177.156.2 e o outro no 220.177.156.3. Tanto faz se cada endereço corresponde a uma placa de rede separada ou se são aliases para uma única placa. O que interessa é que sempre que alguém digitar o endereço IP ou o domínio correspondente no browser será capaz de acessar o site. O IP de cada site é especificado na primeira linha, opção "**VirtualHost**".

A próxima linha "**ServerAdmin**" permite especificar o e-mail do administrador, para onde serão enviadas mensagens de erro e avisos de anormalidades no servidor.

A opção "**DocumentRoot**" é outra configuração crucial, simplesmente porque diz em que pastas ficarão armazenados os arquivos do site em questão. Naturalmente cada site deve ter sua própria pasta, que deve ser acessível ao cliente via ftp, ssh ou outra forma qualquer, para que ele possa dar upload dos arquivos do site.

Isto significa que, além de configurar o Apache, você deve criar para ele um usuário no sistema e configurar um servidor de FTP ou SSH. Para finalizar, use o comando "**chown -R usuário pasta**" para transformar o usuário em dono da pasta e o comando "**chmod 755 pasta**" para acertar as permissões de acesso. Isto faz com que o dono tenha controle total e os demais usuários (e visitantes do site) possam apenas ler os arquivos e executar scripts postos no servidor (como scripts em CGI ou páginas PHP), sem permissão para gravar ou alterar nada.

A opção "**ServerName**" indica o nome de domínio do servidor e não é necessária quando o site é acessado apenas através do endereço IP. Finalmente temos a localização dos dois arquivos de log: **ErrorLog** e **TransferLog**. Por padrão, estes arquivos devem ficar dentro da pasta logs, no diretório raiz do site, separados dos arquivos disponibilizados ao público, que ficam na pasta **www**. Naturalmente você pode usar outras localizações se quiser, é apenas uma convenção.

Name-Based: Esta segunda opção é bem mais usada que a **IP-Based**, por isso deixei por último, caso contrário era capaz de você pular o outro tópico ;-). A configuração baseada em nomes permite que você hospede vários sites, cada um com seu próprio nome de domínio num servidor com um único link e um único IP.

A configuração no arquivo **httpd.conf** é até mais simples que a baseada em IP. A seção fica:

```
NameVirtualHost 192.168.1.100
```

```
ServerName www.kurumin.com.br  
DocumentRoot /var/www/kurumin
```

```
ServerName www.kacique.com.br  
DocumentRoot /var/www/kacique
```

ServerName www.morimoto.com.br
DocumentRoot /var/www/morimoto

A primeira linha ("**NameVirtualHost ***") especifica o endereço IP e a porta do servidor principal. Aqui estou usando como exemplo o endereço "192.168.1.100", pois estou configurando um servidor que vai ficar disponível apenas dentro da rede local. Ao configurar um servidor público, você usaria o endereço IP do servidor na internet, como por exemplo "215.23.45.143".

Em seguida temos as seções **VirtualHost**, que especificam o nome de domínio e o diretório local onde ficam os arquivos de cada um. 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. No registro do domínio, você deverá fornecer dois endereços de DNS (primário e secundário). Se você tiver apenas um, existe um pequeno truque: conecte-se via modem na hora de fazer o registro, assim você terá dois endereços (o do link e o do modem) e conseguirá fazer o registro. Naturalmente, neste caso, você perde a redundância; se o seu link principal cair seu site ficará fora do ar.

É aqui que acaba o trabalho deles e começa o seu. Ao acessar o domínio, o visitante é direcionado para o endereço de DNS fornecido no registro. Isto significa que... bingo! além do Apache você vai precisar de um servidor de **DNS** :-)

O DNS não precisa necessariamente ser uma máquina separada. Mais adiante veremos como instalar e configurar o **Bind** para esta função. Neste caso a requisição do cliente é direcionada da Fapesp para o seu servidor DNS e dele para o servidor Apache. O ciclo se fecha e o cliente consegue finalmente acessar a página.

Se seu servidor estiver hospedando subdomínios, ou seja, endereços como "www.fulano.guiadohardware.net", "www.ciclano.guiadohardware.net", etc., como fazem serviços como o hpg, a configuração continua basicamente a mesma. Você especifica o sub-domínio do cliente na configuração do **VirtualHost** do Apache e também no servidor de DNS.

Uma observação importante é que para o Apache, o domínio "www.fulano.guiadohardware.net" é diferente de apenas "fulano.guiadohardware.net". Para que o site possa ser acessado tanto com o www quanto sem ele, é necessário cadastrar os dois endereços, usando um alias para que o segundo leve ao primeiro.

Como no caso anterior, você deve informar o endereço do seu servidor de DNS no registro do domínio. Como os servidores de registro de domínio lêem as URLs de trás para a frente, todos os acessos a subdomínios dentro do guiadohardware.net serão enviados para o seu servidor DNS e daí para o servidor Apache.

Ao usar o Apache 2, a configuração dos hosts virtuais fica ligeiramente diferente. Ao invés de colocar as configurações de todos os sites diretamente no **httpd.conf**, a

configuração é quebrada em vários arquivos individuais (um para cada site) armazenados dentro da pasta `"/etc/apache2/sites-available"`.

Dentro da pasta 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 por exemplo "kurumin.com.br"), contendo o seguinte:

```
ServerName www.kurumin.com.br
ServerAlias kurumin.com.br
DocumentRoot /var/www/html/kurumin/www/
```

Note que adicionei uma nova diretiva, a "**ServerAlias**", que permite que o site seja acessado tanto com, quanto sem o "www". O DocumentRoot também foi alterado, com relação aos exemplos anterior para refletir a organização padrão adotada no Apache 2, onde os arquivos são por padrão armazenados dentro da pasta `/var/www/html`, ao invés de simplesmente `/var/www`.

Para adicionar mais sites, basta criar vários arquivos, um para cada um, não esquecendo de criar as pastas default de cada.

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 para cada um, dentro da pasta `"/etc/apache2/sites-enabled"`. São estes links que determinam se o site vai realmente ficar disponível. Esta configuração foi adotada para facilitar a vida de quem administra servidores compartilhados por muitos sites. Para desativar um site por falta de pagamento, por exemplo, basta remover o link e recriá-lo quando a situação for normalizada.

Para criar o link para o site "kurumin.com.br" que criamos no exemplo anterior, o comando seria:

```
# cd /etc/apache2/sites-enabled
# ln -s ../sites-available/kurumin.com.br kurumin.com.br
```

Não se esqueça de reiniciar o Apache no final do processo para que as mudanças entrem em vigor.

Esta configuração manual funciona para pequenos servidores, que hospedam algumas dezenas ou centenas de páginas. Grandes serviços de hospedagem geralmente acabam desenvolvendo algum tipo de sistema para automatizar a tarefa. Nos serviços de hospedagem gratuita, por exemplo, onde o número de clientes é assustadoramente grande, as alterações são feitas de forma automática quando o visitante faz seu cadastro, geralmente através de um sistema escrito em PHP ou Java.

Conforme o número de usuários cresce e o espaço em disco no servidor começa a ficar escasso, você começará a sentir falta de um sistema de quotas que limite o espaço que cada usuário pode usar. Para isso, consulte o tópico sobre quotas de disco mais adiante.

Gerando estatísticas

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: procure pelo pacote "**webalizer**" dentro do gerenciador de pacotes. Ele é incluído em todas as principais distribuições. Nas derivadas do Debian, você pode instalá-lo via apt-get: 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 (assim como o **Sarg**). Basta chamá-lo 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/**".

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
```

Para não precisar executar o comando "webalizer" manualmente toda hora, você pode configurar o cron para executá-lo automaticamente uma vez por dia ou uma vez por hora. Para isto, basta criar um script dentro da pasta "**/etc/cron.daily/**" ou "**/etc/cron.hourly/**", contendo o comando "webalizer". Todos os scripts colocados dentro destas pastas são respectivamente executados todos os dias de manhã, ou uma vez por hora. Para que funcione, é importante verificar se o serviço "**cron**" ou "**crond**" está ativo.

O **Mandriva** não inclui o pacote do Webalizer. Para instalar nele, baixe o arquivo "**webalizer-2.xx-xx-static.gz**" no <ftp://ftp.mrunix.net/pub/webalizer/>.

Este arquivo contém um executável genérico, compilado com o objetivo de ser compatível com qualquer distribuição. Todas as bibliotecas necessárias estão incluídas dentro do próprio executável, daí o "**static**" no nome. Para instalá-lo, basta descompactar o arquivo e movê-lo para dentro da pasta "**/usr/local/bin/**":

```
$ wget ftp://ftp.mrunix.net/pub/webalizer/webalizer-2.01-10-static.gz  
$ gunzip webalizer-2.01-10-static.gz
```

```
$ su
# mv webalizer-2.01-10-static /usr/local/bin/webalizer
# chmod 755 /usr/local/bin/webalizer
```

Instalando o suporte a PHP

No início, existiam apenas páginas html estáticas, com links atualizados manualmente. Depois surgiram os scripts **CGI**, geralmente escritos em **Perl**, que permitiram criar vários tipos de formulários e automatizar funções. Finalmente, surgiu o **PHP**, adotado rapidamente como a linguagem padrão para criação de todo tipo de página dinâmica, fórum ou gerenciador de conteúdo.

Além da linguagem ser bastante flexível, um script em PHP chega a ser mais de 100 vezes mais rápido que um script CGI equivalente, além de mais seguro. Em resumo, um script CGI é um executável, que precisa ser carregado na memória, executado e descarregado cada vez que é feita uma requisição. No caso do PHP, o interpretador fica carregado continuamente e simplesmente vai executando de forma contínua os comandos recebidos dos scripts incluídos nas páginas.

Para quem programa em Perl, existe a possibilidade de utilizar o mod-perl, instalável através do pacote "**apache-mod-perl**" ou "**libapache2-mod-perl2**". Assim como o PHP, o mod-perl é um módulo do Apache que fica continuamente carregado na memória, executando os scripts Perl de uma forma bem mais rápida e segura que os scripts CGI.

Mas, voltando ao assunto principal, o suporte a PHP é instalado através do pacote "**php4**" (ou "**php3**", caso você prefira usar a versão antiga, ainda disponível em muitas distribuições). Para instalá-lo, basta usar o gerenciador de pacotes da distribuição em uso, como em:

```
# 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
```

Ao utilizar os pacotes do Debian, ele é instalado na pasta "**/usr/lib/apache/1.3**". Para ativá-lo no Apache 1.3, 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. De forma que o Apache saiba o que fazer com elas:

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

Reinicie o apache para que a alteração entre em vigor:

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

A partir daí, o Apache continua exibindo diretamente páginas com extensão .htm ou .html, mas passa a entregar as páginas .php ou .phps ao interpretador php, que faz o processamento necessário e devolve uma página html simples ao Apache, que se encarrega de enviá-la ao cliente.

Estas páginas processadas são "descartáveis": cada vez que um cliente acessa a página, ela é processada novamente, mesmo que as informações não tenham sido alteradas.

Dependendo do número de funções usadas e da complexidade do código, as páginas em PHP podem ser bastante pesadas. Não é incomum que um site com 300.000 pageviews diários (o que significa umas 20 requisições por segundo nos horários de pico) precise de um servidor dedicado, de configuração razoável.

Quase sempre, os sistemas desenvolvidos em PHP utilizam também um banco de dados MySQL ou PostgreSQL. Para utilizá-los, você precisa ter instalados (além do MySQL ou Postgre propriamente ditos) os módulos "**php4-mysql**" e "**php4-pgsql**", que permitem aos scripts em PHP acessarem o banco de dados:

```
# apt-get install php4-mysql
```

ou

```
# apt-get install php4-pgsql
```

Não se esqueça de reiniciar o Apache, para que as alterações entrem em vigor:

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

Para instalar o suporte a PHP no Apache 2, instale o pacote "**libapache2-mod-php4**" ou "**libapache2-mod-php5**", como em:

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

O módulo "libapache2-mod-php4" é instalado dentro da pasta **"/usr/lib/apache2/modules/"** e é ativado de uma forma diferente que no Apache 1.3. 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/"**:

```
# 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, como em:

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

Instalando o MySQL

O **MySQL** é um banco de dados extremamente versátil, usado para os mais diversos fins. Você pode acessar o banco de dados a partir de um script em PHP, através de um aplicativo desenvolvido em C ou C++, ou praticamente qualquer outra linguagem, ou até mesmo através de um Shell Script!

Existem vários livros publicados sobre ele, por isso vou me limitar a falar sobre a instalação e a configuração necessária para utilizá-lo num servidor **LAMP**, em conjunto com o Apache e o PHP.

O primeiro passo é instalar o servidor MySQL propriamente dito:

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

Você pode instalar também os pacotes "**mysql-client**" (o cliente que permite acessar os dados e fazer modificações no banco de dados) e o "**mysql-navigator**" (uma interface gráfica para ele).

Antes de iniciar o serviço, rode o comando "**mysql_install_db**", que cria a base de dados "**mysql**", usada para armazenar informações sobre todas as outras criadas posteriormente, e uma base chamada "**test**", que pode ser usada para testar o servidor:

```
# mysql_install_db
```

O passo seguinte é ativar o servidor:

```
# /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 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 senha**", incluindo a senha desejada diretamente no comando, como em:

```
# mysqladmin -u root password 123456
```

O próximo passo é criar uma base de dados. Você pode instalar vários scripts diferentes (um fórum, um chat e um gestor de conteúdo, por exemplo) no mesmo servidor e, inclusive, várias cópias de cada um. Isso é cada vez mais utilizado, tanto dentro de sites

que oferecem diversos serviços, quanto em servidores compartilhados, onde os responsáveis por cada site têm a liberdade de instalar os sistemas de sua preferência.

Existem muitas interfaces de administração para o MySQL, mas a forma mais elementar é usar o prompt de comando. Para acessar o prompt do MySQL, use o comando :

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

Enter password:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 43 to server version: 4.0.15-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Veja que o cabeçalho normal do bash foi substituído por um "**mysql>**", que lembra onde você está ;-). Para sair, pressione "**Ctrl+C**" ou execute o comando "**Bye**".

Dentro do prompt, use o comando "**create database**" (criar base de dados), seguido pelo nome desejado. Neste exemplo, estou criando uma base de dados para usar na instalação do phpBB, que veremos a seguir. Um detalhe importante é que todos os comandos dados dentro do prompt do MySQL devem terminar com ponto-e-vírgula:

```
mysql> CREATE DATABASE phpbb;
Query OK, 1 row affected (0.04 sec)
```

Nada impede que você sempre utilize a conta "root" do MySQL e inclusive configure os scripts instalados para o utilizarem. Mas, isto é extremamente inseguro, principalmente se você pretende instalar vários scripts e aplicativos no mesmo servidor, ou se as bases de dados serão acessadas por vários usuários.

O ideal é que cada base de dados tenha um usuário próprio e seja acessível apenas por ele. Se você vai instalar o **phpBB** (fórum) e o **Xoops** (gerenciador de conteúdo), por exemplo, crie duas bases de dados (phpbb e xoops, por exemplo) e dois usuários separados, cada um com permissão para acessar uma das duas bases.

Na configuração de cada um, informe a base de dados a ser usada e o usuário/senha correspondente. Isso evita que um problema de segurança em um coloque em risco também os dados referente ao outro.

Outra situação comum é ao configurar um servidor com vários virtual hosts. Neste caso, o webmaster de cada site vai precisar de uma ou mais bases de dados e, naturalmente, cada um vai ter que ter um login próprio, com acesso apenas às suas próprias bases de dados.

Para criar um usuário "phpbb", com senha "12345" e dar a ele acesso à base de dados "phpbb" que criamos, use (dentro do prompt do MySQL) o comando:

```
mysql> grant all on phpbb.* to phpbb identified by '12345';
```

(permita tudo na base phpbb para o usuário phpbb, identificado pela senha 12345)

Para trocar a senha posteriormente, use o comando:

```
mysql> SET PASSWORD FOR phpbb = PASSWORD('123456');
```

(defina senha para o usuário phpbb, onde a senha é 123456)

Este mesmo comando pode ser usado para trocar a senha do root, como em:

```
mysql> SET PASSWORD FOR root = PASSWORD('asdfgh');
```

Se mais tarde você precisar remover as permissões de acesso de um usuário anteriormente criado (num site com vários webmasters, onde um se desligou da equipe, por exemplo) use o comando:

```
mysql> REVOKE ALL ON phpbb.* FROM phpbb;
```

(remova todos os direitos para a base phpbb, para o usuário phpbb)

Veja que os comandos usados dentro do prompt do MySQL seguem uma linguagem literal, usando palavras do inglês. Quem tem uma boa familiaridade com a língua tem bem mais facilidade em dominar os comandos.

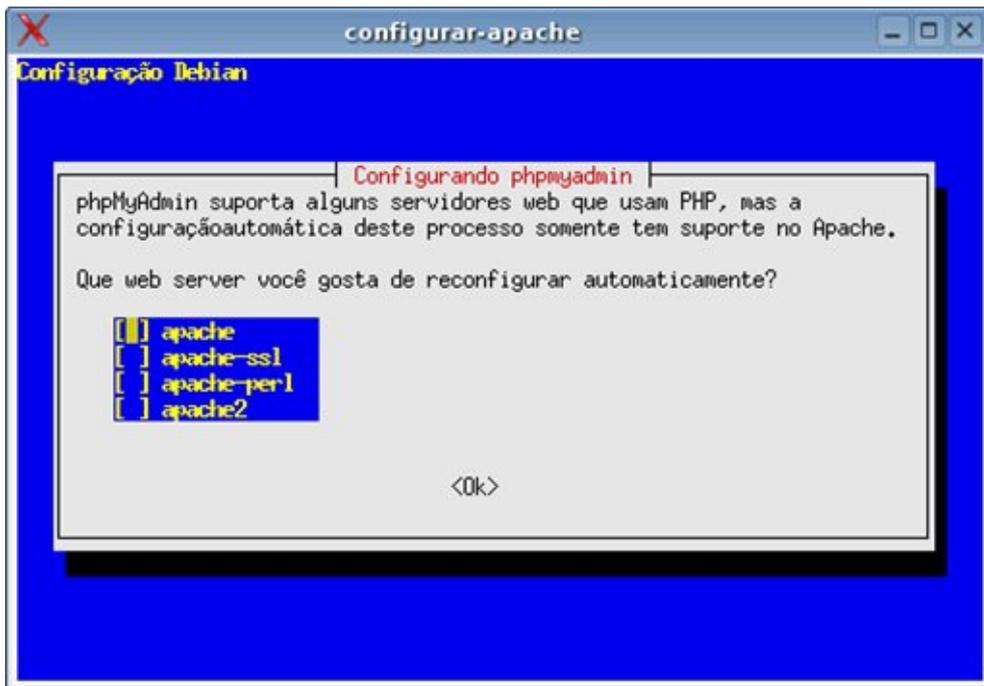
Depois desta configuração inicial, você pode experimentar instalar um gerenciador gráfico para facilitar a manutenção do seu servidor MySQL. Uma boa opção neste caso é o **phpMyAdmin**.

Para instalá-lo, basta instalar o pacote "**phpmyadmin**", como em:

```
# apt-get install phpmyadmin
```

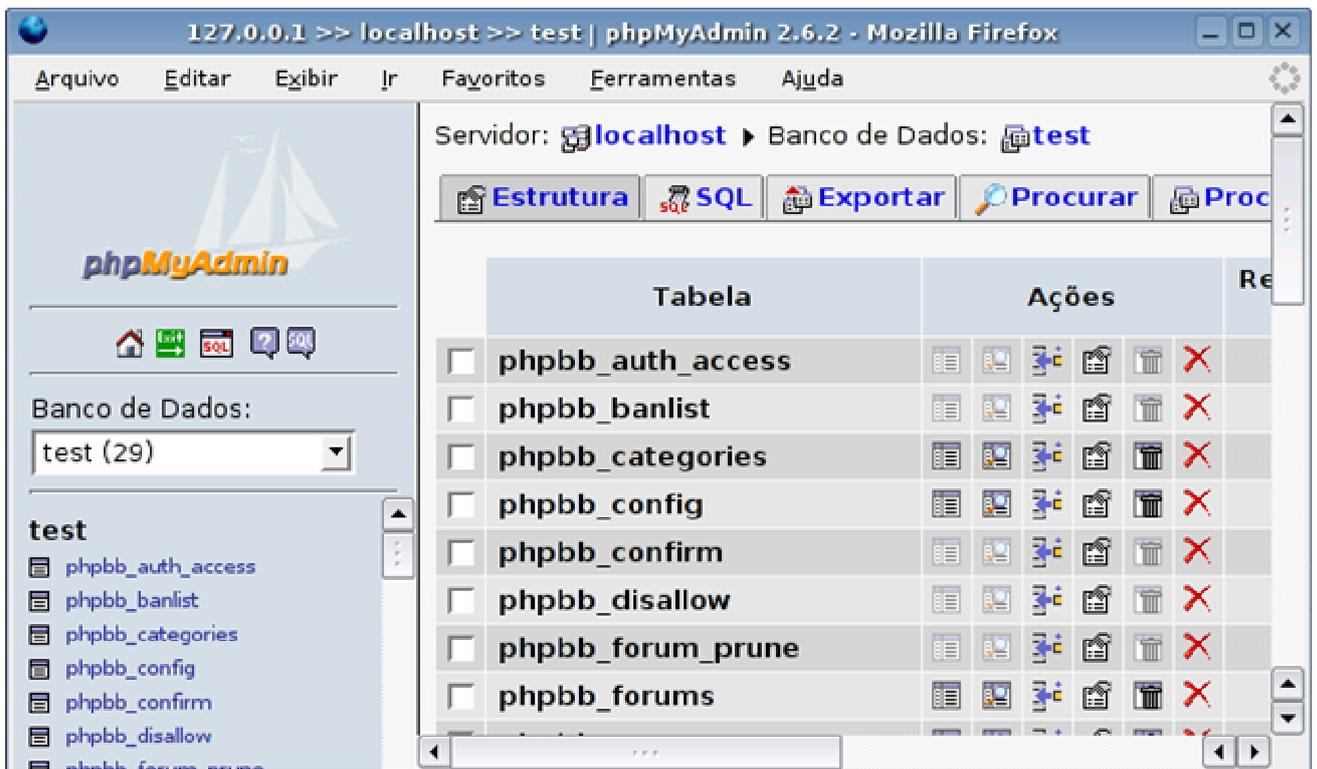
O pacote para instalação em outras distribuições, que não incluam o pacote por padrão, pode ser encontrado no: <http://www.phpmyadmin.net/>.

O phpMyAdmin 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, quando for necessário ter acesso a todas as bases ou fazer backup de tudo, e uma das contas restritas para acessar uma base específica.

Por questões de segurança, a configuração padrão permite que ele seja acessado apenas localmente.



Uma observação importante é que ao ser usado em conjunto com o Apache, instalado no mesmo servidor que ele, o MySQL é acessado apenas localmente, através da interface de loopback. O Apache envia a requisição ao módulo PHP que faz o acesso ao banco de dados, tudo localmente. Nesta configuração o servidor MySQL não deve ficar disponível para a Internet. Configure o firewall para bloquear a porta 3306 usada pelo servidor MySQL, além de todas as outras portas que não forem explicitamente necessárias.

Caso o servidor MySQL for ser utilizado por outros servidores (você pode configurar o phpBB e outros scripts para utilizarem um servidor MySQL externo), deixe a porta aberta apenas para os endereços IP dos servidores que forem ter acesso. Como os servidores dedicados sempre utilizam endereços fixos, ao contrário dos servidores domésticos, esta configuração fica mais simples. Para administrar seu servidor MySQL remotamente, o ideal é que se conecte ao servidor via SSH e faça todo o trabalho através dele. Se precisar acessar diretamente alguma ferramenta de configuração, como o Webmin ou o PhPMyAdmin você pode criar um túnel (novamente usando o SSH) ligando a porta correspondente do servidor à uma porta da sua máquina e fazer o acesso através dela.

Instalando o phpBB

Com o **php4**, **php4-mysql** e o **mysql-server** instalados você tem pronta a estrutura necessária para instalar os diversos scripts de fórum, chat, gestores de conteúdo e outros.

A maioria destes scripts é simples de instalar. Você precisa apenas criar uma base de dados no **MySQL** ou **Postgre**, copiar os arquivos para uma pasta dentro do servidor web e editar um arquivo (ou acessar uma página de configuração através do navegador) para incluir as informações sobre o servidor (base de dados a ser usada, login e senha, etc.) e concluir a configuração.

Note que embora o Apache e o MySQL sejam bastante seguros, nada garante que os scripts desenvolvidos por terceiros também serão. De nada adianta ter um servidor web extremamente seguro, se o script de gerenciamento de conteúdo que você instalou tem um buffer overflow no campo de login, que permite executar comandos arbitrários, o que pode ser usado para obter a senha do servidor MySQL (que o script usa para fazer seu trabalho) e, de posse dela, fazer alterações no conteúdo do site.

O ponto fraco na segurança de qualquer site ou fórum é quase sempre a segurança do script usado. Não escolha qual usar pensando apenas na facilidade de uso. Investigue o histórico de segurança e, uma vez escolhido qual usar, fique de olho nas atualizações de segurança.

Vou usar como exemplo a instalação do phpBB, um script de fórum bastante usado, e com um bom histórico de segurança e desempenho. Ele é o script usado no fórum do Guia do Hardware (<http://forumgdh.net/>), que tem um milhão e meio de mensagens postadas, 30 mil usuários registrados e chega a ter 400 acessos simultâneos.

O phpBB tem código aberto e é gratuito, você pode baixá-lo no:
<http://www.phpbb.com/downloads.php>

Comece baixando o pacote principal. Enquanto escrevo, ele está na versão 2.0.15 e o arquivo é o "**phpBB-2.0.15.tar.gz**".

Para instalar, salve-o dentro da pasta **"/var/www"** (ou a pasta de dados do seu servidor, caso esteja usando outro diretório) e renomeie a pasta criada para o diretório onde o fórum deve ficar acessível. No meu caso estou instalando na pasta "forum/". Delete o arquivo original, pois não vamos mais precisar dele:

```
# cd /var/www
# tar -zxvf phpBB-2.0.15.tar.gz
# rm -f phpBB-2.0.15.tar.gz
# mv phpBB2/ forum/
```

Aproveite para instalar também o suporte a internacionalização. O phpBB já foi traduzido para várias linguagens, incluindo português do Brasil.

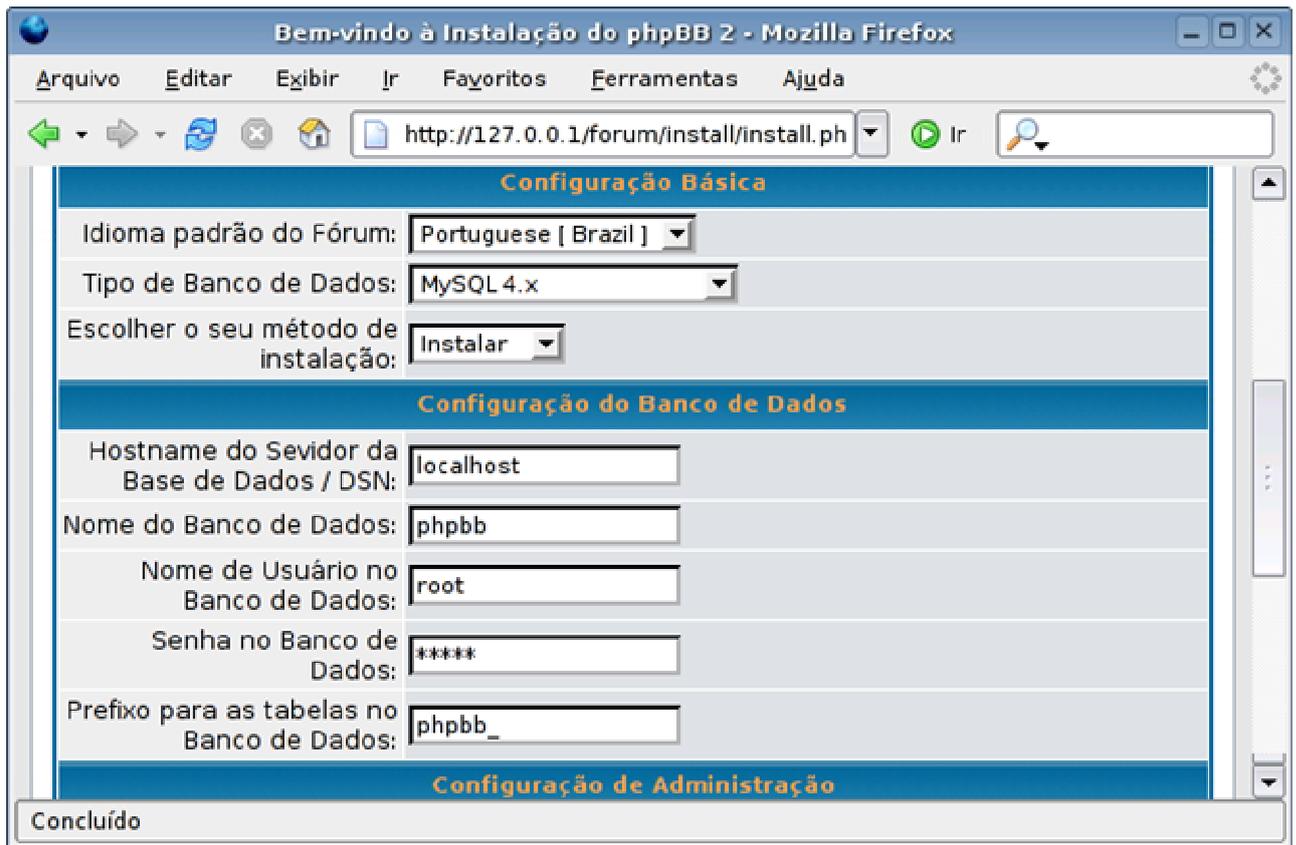
Comece baixando o arquivo **"lang_portuguese.tar.gz"** (que contém a tradução propriamente dita) e descompacte-o dentro da pasta **"/var/www/forum/language"**.

Baixe agora o arquivo **"subSilver_portuguese_brazil.tar.gz"** (que contém botões e ícones com o texto de legenda traduzido) e descompacte-o na pasta **"/var/www/forum/templates"**.

Veja que tudo isto pode ser feito via ftp ou sftp, mesmo que você não tenha acesso via shell no servidor. Tudo que é preciso fazer é copiar os arquivos para as pastas apropriadas. Este sistema de instalação foi desenvolvido pensando em quem utiliza planos de hospedagem em servidores compartilhados.

Depois de copiar os arquivos, acesse a página **"/forum/install/install.php"** dentro da árvore do seu site. O acesso pode ser feito tanto localmente (**http://127.0.0.1/forum/install/install.php**) quanto via internet. Esta é a página usada para concluir a instalação.

Note que você precisa acessar a página assim que os arquivos forem copiados, pois ela fica acessível para qualquer um.



Preencha os campos com as informações do seu servidor:

- **Database Type:** Escolha o MySQL 4.x ou MySQL 3.x, de acordo com o a versão instalada. Note que o phpBB também oferece suporte ao PostgreSQL e até mesmo ao MS SQL Server, caso o Apache esteja rodando sobre o Windows.
- **Database Server Hostname / DSN:** O phpBB pode acessar um servidor MySQL instalado em outra máquina da rede, não é necessário que o Apache e o MySQL estejam instalados na mesma máquina. Separar os dois é interessante do ponto de vista da performance e também da segurança, mas por outro lado é mais caro e trabalhoso. Caso o MySQL esteja instalado na mesma máquina, mantenha o "localhost", do contrário, informe o endereço IP ou domínio do servidor a ser utilizado.
- **Your Database Name:** Aqui você indica a base de dados que será usada pelo fórum. No tópico anterior, criamos a base de dados "phpbb". Caso você esteja instalando só para testar, pode usar também a base de dados "test", criada por padrão.
- **Database Username:** Ao criar a base "phpbb" criamos também o usuário "phpbb", com acesso a ela. Ao instalar para teste, você pode usar a conta "root", que tem acesso a tudo, mas isso não é recomendável do ponto de vista da segurança. Nunca use o root numa instalação que vá ficar disponível na internet.
- **Database Password:** A senha do usuário indicado na opção acima.

- **Prefix for tables in database:** phpbb_ (mantenha o default).
- **Admin Email Address:** seu@mail.com (um e-mail válido para o envio de mensagens de erro e alertas).
- **Domain Name:** Aqui vai o domínio do seu site, como "meunome.com.br". Se você está fazendo uma instalação de teste, fora do servidor real, deixe o valor padrão.
- **Server Port:** 80 (porta onde o servidor Apache está disponível. Informe a porta correta caso tenha alterado a configuração do Apache)
- **Script path:** /forum/ (pasta no servidor onde está instalado o fórum).
- **Administrator Username:** admin (o login que você usará para administrar o fórum).

Ao terminar, clique no "**Start Install**" e feche a janela do navegador.

Caso apareçam mensagens de erro significa que o servidor Apache ou o MySQL não estão instalados corretamente. Verifique se todos os pacotes foram instalados sem erros, se o servidor MySQL está ativo e se você não se esqueceu de reiniciar o Apache depois de ter instalado o suporte a PHP.

É importante deletar as pastas "**install**" e "**contrib**" dentro da pasta do fórum, que contém arquivos necessários apenas durante a instalação:

```
# cd /var/www/forum/  
# rm -rf install contrib
```

Terminados estes passos, seu fórum já estará funcionando. Assim como qualquer gerenciador que se preze, o phpBB oferece um painel de administração, que fica disponível ao se logar usando a conta administrativa criada durante a instalação. Através do painel, você pode criar novas salas, alterar as configurações do fórum, moderar e assim por diante.

yourdomain.com :: Índice - Mozilla Firefox <2>

Arquivo Editar Exibir Ir Favoritos Ferramentas Ajuda

http://127.0.0.1/forum/ Ir

Quem está ligado

Os nossos Usuários colocaram um total de **1** mensagens
Temos **1** Usuários registrados
Dêem boas vindas ao nosso mais novo usuário: [admin](#)

 Há **1** Usuário online :: Nenhum Usuários Registrado, Nenhum Invisível e 1 Visitante [[Administrador](#)] [[Moderador](#)]
Recorde de Usuários Online foi de **1** em Qua Mai 18, 2005 12:44 am
Usuários Registrados Nenhum

Esta informação é baseada em Usuários Ativos nos últimos cinco minutos

Entrar

Usuário: Senha: Permanecer Logado automaticamente em cada visita

 Mensagens novas  Não há mensagens novas  Fórum Bloqueado

Concluído

Configurando um servidor de e-mails

Tradicionalmente, o **Sendmail** é o servidor de e-mails mais conhecido, não apenas no Linux, mas nos sistemas Unix em geral. Ele é um dos mais antigos (disponível desde 1982, mais de uma década antes da popularização da Internet) e foi a opção padrão de 9 em cada 10 administradores de sistemas durante muito tempo.

Apesar disso, o uso do Sendmail vem decaindo de forma estável de uma década para cá. As queixas podem ser resumidas a duas questões fundamentais. A primeira é o brutal número de opções e recursos disponíveis, que tornam a configuração bastante complexa e trabalhosa. Muitos administradores da velha guarda gostam da complexidade, mas a menos que você pretenda dedicar sua vida à arte de manter servidores Sendmail, ela acaba sendo um grande problema.

A segunda questão é o histórico de vulnerabilidades do Sendmail que, na melhor das hipóteses, pode ser definido como "muito ruim". É verdade que nos últimos anos as coisas melhoraram bastante, mas as cicatrizes do passado ainda incomodam.

O concorrente mais antigo do Sendmail é o **Exim**, que oferece um conjunto bastante equilibrado de recursos, boa performance e um bom histórico de segurança. O EXIM é o MTA usado por padrão no Debian, ele é instalado automaticamente como dependência ao instalar pacotes que necessitem de um servidor de e-mails, mas pode ser rapidamente substituído pelo Postfix ou o Sendmail via apt-get, caso desejado.

O **Qmail** é uma escolha mais complicada. Quando foi lançado, em 1997, o Qmail trouxe várias inovações e um design bastante simples e limpo, com ênfase na segurança, o que o tornou rapidamente uma opção bastante popular.

Entretanto, o Qmail possui dois graves problemas. Ele foi abandonado pelo autor em 1998, depois do lançamento da versão 1.03 e, embora o código fonte seja aberto, a licença de uso impede a redistribuição de versões modificadas, embora seja permitido disponibilizar patches.

Ao longo dos anos, surgiram várias iniciativas de atualizações do Qmail, onde o código original é distribuído junto com um conjunto de patches com atualizações. Para instalar, você precisa primeiro aplicar cada um dos patches, para em seguida poder compilar e instalar o Qmail. Dois dos projetos mais populares são o <http://qmail.org/netqmail/> e o <http://www.qmailrocks.org/>.

Embora o Qmail ainda possua uma legião de seguidores fiéis, a limitação imposta pela licença acaba sendo um grande empecilho para quem deseja utilizá-lo e representa uma grande ameaça à manteneabilidade dos patches a longo prazo, já que as alterações em relação ao código original tornam-se cada vez mais complexas e difíceis de aplicar, com a disponibilização de patches para patches que já são patches para outros patches.. ;).

Finalmente, temos o **Postfix**. Ele é uma espécie de meio termo entre a simplicidade do Qmail e a fartura de recursos do Exim. Entre os três, ele é o mais rápido e o mais simples de configurar, o que faz com que ele seja atualmente o mais popular e o que possui mais documentação disponível. O Postfix também possui um excelente histórico de segurança, sendo considerado por muitos tão seguro quanto o Qmail.

Existem fortes motivos para não usar o Sendmail ou o Qmail em novas instalações, mas temos uma boa briga entre o Postfix e o Exim. Escolhi abordar o Postfix aqui simplesmente por que, entre os dois, ele é mais popular, o que torna mais simples encontrar documentação e conseguir ajuda quando tiver dúvidas.

Apesar disso, a maior parte dos conceitos podem ser usados também na configuração do Sendmail e outros servidores; afinal, a configuração de todos eles reserva mais semelhanças que diferenças.

Instalando o Postfix

O pacote do Postfix pode ser encontrado em todas as principais distribuições. Nas distribuições derivadas do Debian, você pode instalá-lo usando o apt-get:

```
# apt-get install postfix
```

Mais três pacotes que adicionam algumas funcionalidades importantes são:

```
# apt-get install postfix-ldap
```

(permite configurar o servidor para obter a lista de logins e senhas a partir de um servidor LDAP)

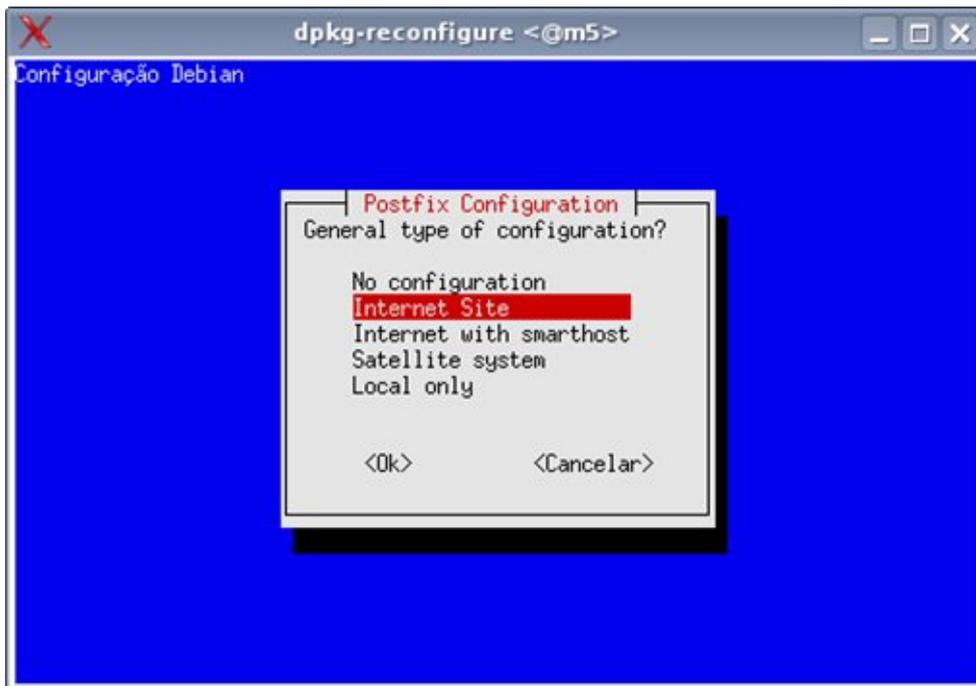
```
# apt-get install postfix-mysql
```

```
# apt-get install postfix-pgsql
```

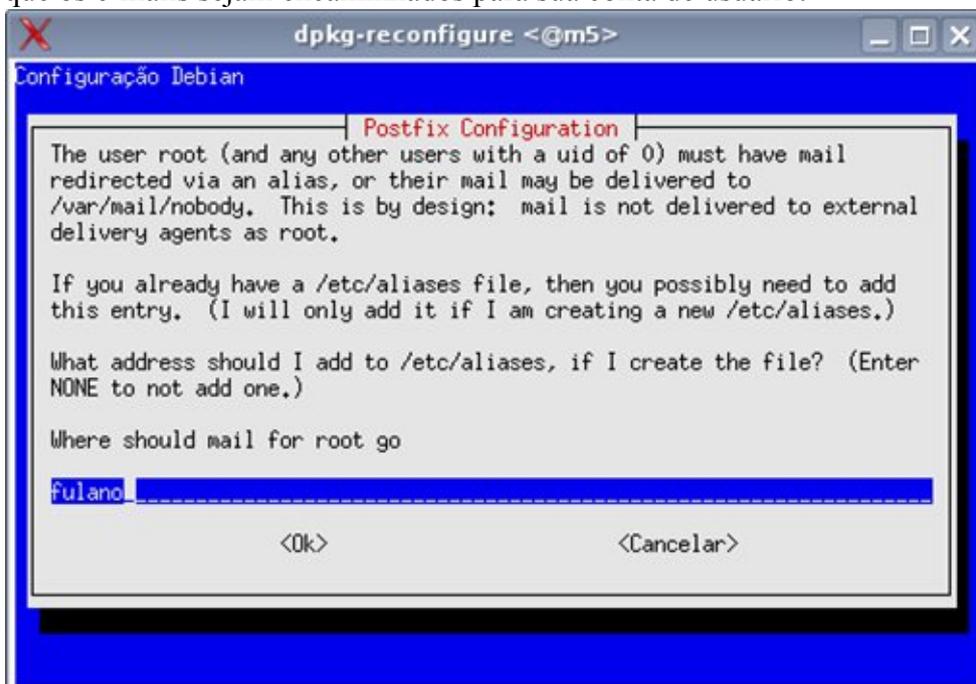
(para usar um servidor MySQL ou Postgree para armazenar a lista de logins e senhas)

O pacote do Debian possui um wizard configuração, exibido durante a instalação do pacote. Ele faz algumas perguntas e se encarrega de gerar uma configuração básica, suficiente para colocar o servidor para funcionar. Ele não faz nada de sobrenatural, apenas ajusta o `/etc/postfix/main.cf` de acordo com as opções definidas. Por enquanto, vou apenas explicar rapidamente as opções, pois as veremos com mais detalhes ao estudar a configuração manual do postfix.

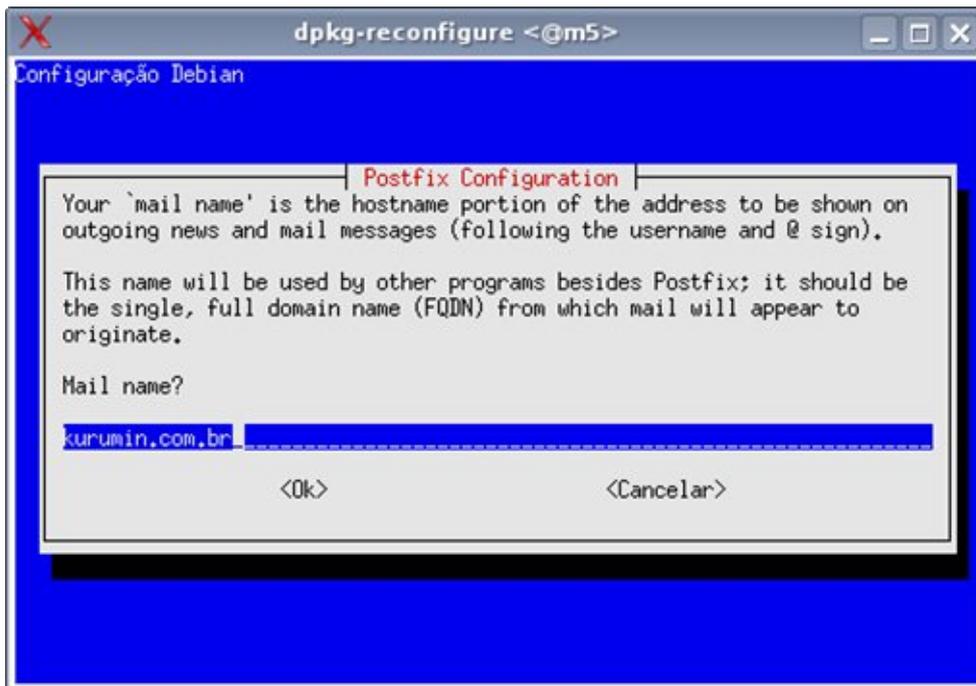
A primeira pergunta é sobre a função do servidor de e-mails que você está configurando. A opção mais usada é "Internet Site", onde você cria um servidor "de verdade", que envia e recebe os e-mails diretamente. Na segunda opção "with smarthost" seu servidor recebe mensagens, mas o envio fica a cargo de outra máquina, enquanto na terceira ("Satellite system", a mais limitada) seu servidor envia através de outra máquina e não recebe mensagens. A opção "Local only" é usada apenas em redes de terminais leves (poderia ter alguma utilidade num servidor LTSP, por exemplo), pois permite apenas que os usuários logados no servidor troquem e-mails entre si.



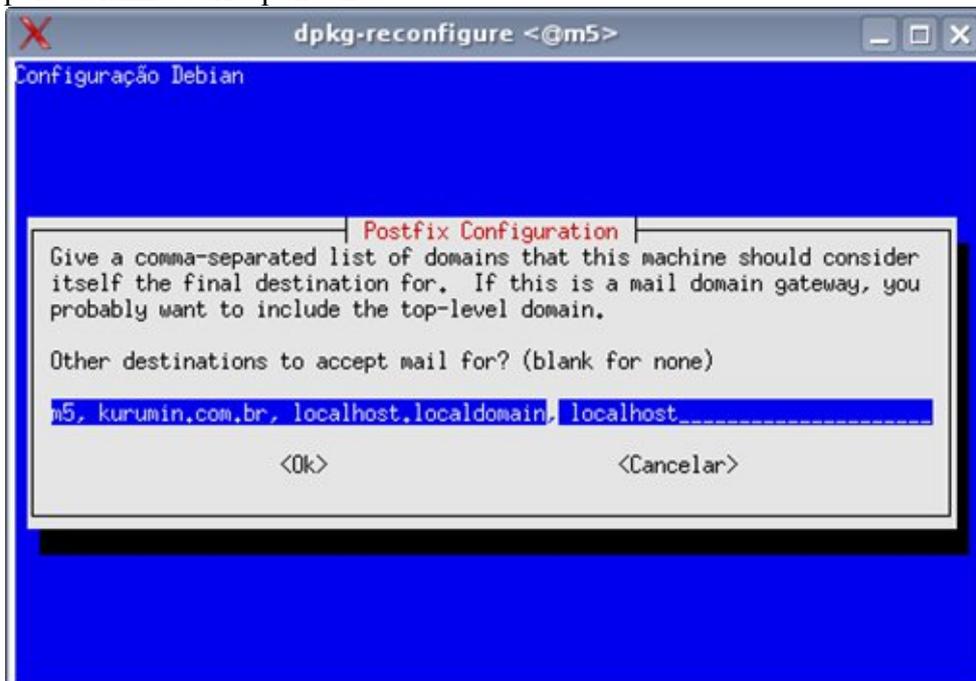
Nos sistemas Linux, é recomendado que você use a conta root apenas para a manutenção do sistema. Mesmo sendo o administrador, você usa uma conta normal de usuário, utilizando o su ou sudo para ganhar privilégios de root apenas quando necessário. Se você quase nunca usa a conta root, significaria que os e-mails enviados para o "root@seu-servidor" nunca seriam lidos. A segunda pergunta mata a questão, permitindo que os e-mails sejam encaminhados para sua conta de usuário:



A terceira pergunta é sobre o domínio do servidor, que será incluído nas mensagens enviadas. Se o você está configurando um servidor dedicado use seu domínio registrado. Se está apenas configurando um servidor de testes, pode usar o nome da máquina:

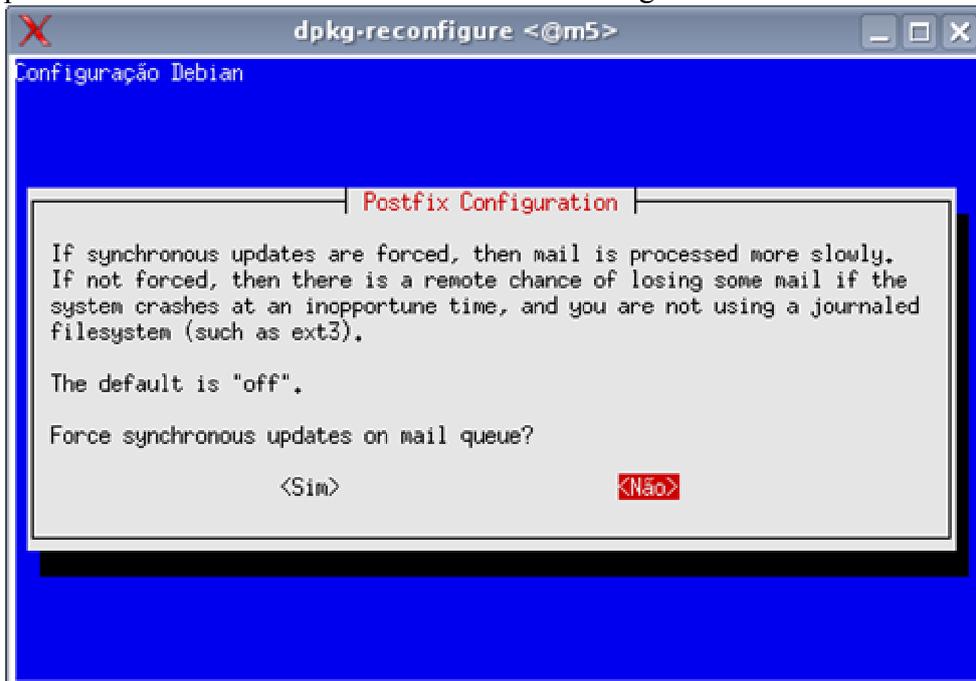


A questão seguinte já é um pouco mais complexa. Você deve definir os destinos que serão aceitos pelo seu servidor, ou seja, os endereços que colocados no destinatário da mensagem fazem ele entender que o e-mail é para ele. Aqui você usa o nome da máquina, o domínio registrado (no caso de um servidor real), "localhost.localdomain" e "localhost", todos separados por vírgula e espaço. esta forma, qualquer e-mail destinado ao "fulano@kurumin.com.br", "fulano@m5" (o nome da máquina) ou "fulano@localhost", que chegue até seu servidor, será encaminhado para a caixa postal do usuário "fulano". Em compensação, um e-mail destinado ao "ciclano@guiadohardware.net", por exemplo, será repassado ao servidor responsável pelo domínio correspondente.



A opção "synchronous updates" permite desativar as otimizações no envio das mensagens, fazendo com que os e-mails sejam enviados conforme são recebidos e em ordem. Esta opção aumenta um pouco a confiabilidade do servidor, pois reduz a

possibilidade de perda de mensagens ainda não enviadas, em casos de travamentos ou quedas de energia. Por outro lado, ela reduz substancialmente o desempenho do servidor, por isso nunca deve ser ativada em servidores de grande volume.



Depois de concluída a instalação, o servidor já estará iniciado e configurado para subir automaticamente durante o boot. Em algumas distribuições, como no Mandriva, o servidor é configurado para subir durante o boot, mas não fica ativado depois da instalação, para que você tenha a chance de revisar o arquivo de configuração antes de ativá-lo. Neste caso, você precisa iniciar o servidor manualmente usando o comando "service postfix start", ou "/etc/init.d postfix start".

O servidor SMTP escuta, por padrão, na porta 25. Os e-mails são transmitidos de uma forma bem simples, com comandos de texto. Uma forma de entender como isso funciona é mandar um e-mail interno para o root do sistema, usando o telnet.

Sim, os servidores SMTP podem ser acessados via telnet, basta mandar o cliente se conectar na porta 25. Isso permitirá enviar o e-mail de testes conversando direto com o servidor Postfix. Se o IP do servidor na rede interna for 192.168.1.33, por exemplo, o comando seria:

```
$ telnet 192.168.1.33 25
```

```
Trying 192.168.1.33...
Connected to 192.168.1.33.
Escape character is '^]'.
220 kurumin ESMTP Postfix (Debian/GNU)
HELO smtp.eu.com
250 kurumin
MAIL From: eu@eu-mesmo.com
250 Ok
RCPT to: joao@localhost
250 Ok
DATA
```

354 End data with <CR><LF>.<CR><LF>

Vai ver se estou na esquina!

.

250 Ok: queued as 8CEDB2215

QUIT

221 Bye

Connection closed by foreign host.

As linhas em negrito são os comandos executados no terminal, seguidos pelas respostas do servidor. O comando "HELO" serve para iniciar a conversa, onde o emissor se identifica. Os passos seguintes são dizer o emissor do e-mail (MAIL From:) e o destinatário (RCPT to:), seguido pelo texto do e-mail (DATA). Note que depois do texto vem uma linha com um ponto, que indica o final da mensagem.

No caso, enviei um mail com remetente falso para o usuário "joao" da máquina (joao@localhost). Este e-mail local pode ser lido usando um cliente de modo texto, como o mutt:

```
# apt-get install mutt
```

Da próxima vez que você se logar com o usuário especificado, verá uma mensagem avisando da polida mensagem que foi enviada:

You have new mail.

Chamando o mutt, você verá que e-mail realmente está lá.

Antigamente, antes da popularização da internet, esses e-mails locais eram comuns, pois geralmente várias pessoas usavam o mesmo servidor (e cada servidor possuía vários terminais burros ligados a ele). As mensagens eram trocadas diretamente entre os servidores e armazenadas no spool. Quando o usuário se logava, tinha acesso à sua caixa postal.

Hoje em dia, pouca gente ainda utiliza o mutt. Em geral usamos servidores POP3 ou IMAP para armazenar as mensagens e as baixamos de vez em quando usando algum cliente de e-mails gráfico. A idéia continua sendo basicamente a mesma, mas agora em escala muito maior. Cada e-mail enviado passa por vários servidores antes de chegar ao destinatário, as mensagens são armazenadas no servidor POP3 ou IMAP do servidor e, quando o destinatário se conecta, baixa todas as mensagens de uma vez.

O Postfix (ou Qmail ou Sendmail) armazena as mensagens em uma pasta local, por padrão a pasta "Maildir", dentro do diretório home de cada usuário. Programas como o Mutt acessam diretamente as mensagens dentro da pasta, mas, para baixar as mensagens remotamente, via pop3 ou imap, você precisa instalar um componente adicional.

Existem vários servidores pop3, como o Cyrus e o Courier. O Courier é o mais usado, pois inclui vários componentes adicionais. Ele é, na verdade, uma suíte, que inclui até mesmo um webmail.

Para instalar o módulo pop3, instale o pacote:

```
# apt-get install courier-pop
```

Aproveite para instalar também a encriptação via ssl. Este recurso é importante hoje em dia, pois sem encriptação, seus e-mails (incluindo o login e senha) são transmitidos em texto plano pela rede e podem ser interceptados. Uma vez ativo o recurso no servidor, basta marcar a opção no cliente de e-mails.

```
# apt-get install courier-pop-ssl
```

Para instalar o servidor imap, instale os pacotes:

```
# apt-get install courier-imap  
# apt-get install courier-imap-ssl
```

Com esta configuração básica você já conseguirá enviar e receber e-mails. Inicialmente, você pode testar pedindo para alguém enviar um e-mail para seu endereço IP, como em: fulano@200.220.123.32. Se tudo estiver funcionando, o próximo passo é configurar o servidor DNS (<http://www.guiadohardware.net/tutoriais/120/>) para que você possa receber e-mails através do seu domínio registrado.

Não é uma boa idéia receber e-mails usando uma conexão ADSL, pois uma conexão instável fará com que alguns e-mails sejam perdidos. Outro problema é que quase todas as faixas de endereços de conexões via ADSL nacionais fazem parte de listas negras de spam (justamente por já terem sido exaustivamente usadas para envio de spam no passado). Nesse caso, é melhor configurar seu servidor como um sistema satélite, onde é usado um servidor SMTP externo para envio de e-mails. Você pode usar o próprio SMTP do provedor, ou o servidor de uma empresa de hospedagem, que tenha o nome "limpo" na praça.

De qualquer forma, nada impede que você registre uma conta em um serviço de DNS dinâmico, como o <http://no-ip.com> e experimente manter seu servidor de e-mails para fins didáticos.

Ao usar os pacotes **courier-pop-ssl** ou **courier-imap-ssl**, é necessário gerar um certificado. Empresas como a Verisign vendem certificados reconhecidos, que são necessários caso você queira abrir um site de comércio eletrônico, por exemplo. Mas, para um servidor particular, não existe nada errado em gerar seu próprio certificado. Ele vai funcionar da mesma forma e, se corretamente gerado, com a mesma segurança. O único efeito desagradável é que os clientes receberão uma mensagem "Não é possível comprovar a autenticidade do certificado..." ao se conectarem.

Para criar uma chave para o servidor **IMAP**, comece renomeando a chave de testes, criada durante a instalação:

```
# cd /etc/courier  
# mv imapd.pem imapd.pem.old
```

Edite agora o arquivo "**imap.conf**" (na mesma pasta), colocando as informações sobre o país, cidade, nome do servidor, etc. Depois, basta gerar o novo certificado com o comando:

```
# mkimapdcert
```

Para gerar a chave para o servidor POP3, o procedimento é quase o mesmo. Dentro da pasta "/etc/courier" remova ou renomeie o arquivo "pop3d.pem", edite o arquivo "pop3d.cnf", colocando as informações do servidor e gere o certificado com o comando "mkpop3dcert".

Cadastrando usuários e Configurando

Cadastrando usuários

Lendo a documentação, parece que cadastrar usuários no servidor de e-mails é muito complicado, pois existem muitas formas de fazer isso. A forma mais simples é simplesmente criar um novo usuário no sistema, usando o comando adduser, como em:

```
# adduser joao
```

Desde que o servidor de e-mails esteja instalado, será criada a conta joao@servidor, acessível tanto localmente (usando o mutt e outros clientes), quanto remotamente, via pop3 ou imap.

O problema é que o usuário João passa a poder logar-se na máquina de outras formas, via ssh, telnet, acessar compartilhamentos de rede e assim por diante. Essa abordagem serve para servidores internos, onde os usuários são conhecidos ou funcionários da mesma empresa, mas não é um sistema adequado para um grande servidor web, com inúmeras contas de e-mails de usuários desconhecidos ou que hospeda um servidor Apache com vários subdomínios.

Hoje em dia existem várias outras opções para cadastrar contas no servidor de e-mails, sem precisar necessariamente criar logins válidos no sistema. Você pode armazenar as contas em um servidor MySQL, Postgree SQL ou até mesmo em um servidor LDAP. Para isso, usamos os pacotes **postfix-ldap**, **postfix-mysql** ou **postfix-pgsql**, que vimos anteriormente.

Configurando

Antes de começar a falar sobre a configuração do Postfix, é importante que você entenda alguns termos usados com frequência nos arquivos de configuração e tutoriais:

- **MTA** (Mail Transport Agent): É o servidor de e-mails propriamente dito, com o Postfix, Qmail, Sendmail e o Exim. Um MTA obrigatoriamente suporta enviar e receber e-mails via SMTP, o protocolo utilizado para transportar as mensagens entre os servidores. O servidor pode ser configurado para enviar e receber os e-mails diretamente (internet site) ou se limitar a receber mensagens, usando um servidor SMTP externo (smarthost) na hora de enviar.

Normalmente, você configura seu servidor como "internet site" apenas ao utilizar um servidor dedicado, ou caso sua empresa possua um link dedicado, com um IP "limpo", fora dos cadastros das listas negras de spam (você pode checar através do <http://rbls.org/>).

- **Mua** (Mail user agent): Este é o nome técnico do cliente de e-mail, como o Thunderbird, Evolution, Kmail, etc. usados diretamente pelo usuário final.
- **MDA** (Mail Delivery Agent): O MDA funciona como um intermediário entre o MTA e o Mua. Ele não é obrigatório, mas pode fazer algumas coisas úteis, como aplicar filtros antispam, remover vírus anexados nas mensagens, encaminhar para outros endereços e assim por diante. Dois exemplos usados no Linux são o Fetchmail e o Procmail. Você os utiliza quando precisa baixar as mensagens do provedor e aplicar filtros diversos antes de encaminhá-las aos usuários.

O principal arquivo de configuração do Postfix é o `"/etc/postfix/main.cf"`. Este é um exemplo de arquivo de configuração funcional. Veja que, apesar da complexidade da tarefa, a configuração do Postfix é relativamente simples:

```
# /etc/postfix/main.cf
myhostname = etch.kurumin.com.br
mydomain = kurumin.com.br
append_dot_mydomain = no
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = etch.kurumin.com.br, kurumin.com.br, localhost
relayhost =
mynetworks = 127.0.0.0/8
home_mailbox = Maildir/
mailbox_command =
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
message_size_limit = 20000000
mailbox_size_limit = 0
```

Vamos a uma explicação mais detalhada de cada uma das opções:

Embora não seja citado no arquivo, o postfix roda utilizando uma conta com privilégios limitados, de forma a limitar o dano no caso de qualquer problema de segurança relacionado ao servidor de e-mails). Estas opções já vem configuradas por padrão ao instalar o pacote.

As primeiras linhas do arquivo indicam o nome da máquina e o domínio. Caso seu servidor não tenha um domínio registrado, ou é usado apenas dentro da rede local), você pode usar o "localdomain" como domínio. Note que muitos servidores rejeitam mensagens enviadas por servidores sem domínio registrado, para dificultar o envio de spams. É por isso que é tão importante configurar corretamente o DNS reverso no Bind, já que é através dele que os servidores remotos podem verificar se os e-mails realmente vêm do seu domínio.

A opção "myhostname" deve conter o nome completo do servidor, incluindo o domínio,

enquanto que a opção "mydomain" contém apenas o domínio, sem o nome da máquina, como em:

```
myhostname = etch.kurumin.com.br
mydomain = kurumin.com.br
append_dot_mydomain = no
```

A linha "mydestination" Esta linha indica quais nomes e domínios serão considerados endereços locais pelo servidor. Se o nome do servidor é "kurumin.kurumin.com.br" e o "domínio "kurumin.com.br", o servidor entenderia que tanto e-mails endereçados a "usuario@etch.kurumin.com.br", quanto "usuario@kurumin.com.br" e "usuario@localhost" são endereçados a ele mesmo.

```
mydestination = etch.kurumin.com.br, kurumin.com.br, localhost
```

A linha "mynetworks" especifica os endereços ou faixas de endereços a partir de onde o servidor aceitará o envio de mensagens.

É preciso configurar esta opção com muito cuidado, caso contrário um spammer poderá usar seu servidor para enviar mensagens não solicitadas, consumindo sua banda e possivelmente fazendo seu servidor ser incluído em várias blacklists, o que vai lhe causar muita dor de cabeça.

A opção 'mynetworks = 127.0.0.0/8' permite apenas e-mails enviados localmente. Você pode especificar várias faixas de endereços separando-os com vírgula, como em: "mynetworks = 200.221.149.0/24, 127.0.0.0/8".

```
mynetworks = 127.0.0.0/8
inet_interfaces = all
```

Na opção "relayhost" você pode indicar um servidor SMTP externo, através do qual as mensagens serão enviadas. Deixando esta opção em branco, todos os e-mails serão enviados diretamente pelo seu servidor.

Hoje em dia é bem mais simples usar um servidor externo por causa da questão do spam. Usar o smtp de um provedor conhecido fará com que menos mensagens se percam nos filtros dos destinatários.

Para usar um relayhost aqui, é preciso indicar um servidor que aceite mensagens enviadas por este servidor sem pedir autenticação. Em geral, as empresas que oferecem serviços de hospedagem oferecem esta opção em troca de uma taxa adicional. É possível também configurar seu provedor para se autenticar, com um pouco mais de trabalho.

Ex: relayhost = smtp.meuprovedor.com

Opcionalmente, você pode configurar os clientes de e-mail nas estações para usarem diretamente o smtp do provedor, deixando seu servidor postfix apenas para receber. Nesse caso, você pode usar qualquer smtp a que tenha acesso.

```
relayhost =
```

A linha "home_mailbox" indica a pasta local, dentro do home de cada usuário, onde os e-

mails ficarão armazenados. A pasta Maildir/ é o padrão usado por diversos MTA's. Caso necessário, crie a pasta manualmente, usando o comando "maildirmake ~/Maildir" (executado como o usuário para o qual a pasta será criada). Em seguida, execute o comando "maildirmake /etc/skel/Maildir" como root, para que todos os novos usuários criados a partir daí já venham com a pasta criada. Normalmente, os pacotes instalados pelas distribuições automatizam esta etapa.

```
home_mailbox = Maildir/  
recipient_delimiter = +  
inet_interfaces = all
```

Na maioria dos casos, é desejável limitar o tamanho das mensagens recebidas, para evitar que algum espartinho envie um ISO de CD anexado à mensagem, consumindo toda a banda e acabando com o espaço em disco do servidor. O padrão do postfix é limitar as mensagens a 10 MB. Qualquer anexo maior que isso é recusado. Esta configuração pode ser alterada através da opção "message_size_limit", onde você especifica o valor desejado, em bytes. Note que por causa do uso do MIME, o tamanho dos anexos cresce substancialmente ao serem enviados via e-mail. Um arquivo de 5 MB, transforma-se numa mensagem de quase 7. Leve isto em conta ao definir o limite. Aqui estou usando um limite de 20 MB decimais:

```
message_size_limit = 20000000
```

A opção "mailbox_size_limit" serviria para definir o limite de armazenamento para a caixa postal do usuário. Entretanto, ao usar o formato Maildir para as caixas postais, cada mensagem é salva num arquivo separado, de forma que a opção não funciona. Por isso, usamos o valor "0" para desativá-la. A melhor forma de limitar o espaço dos usuários é simplesmente definir quotas de espaço em disco, usando o Quota.

```
mailbox_size_limit = 0
```

Em geral, os arquivos de configuração padrão, incluídos nas distribuições, são suficientes para ter um servidor de e-mails básico funcional. Mas, depois de feito o primeiro teste, nunca deixe de editar o arquivo, verificando todas as opções. Você pode tanto usar como ponto de partida o arquivo original, quanto usar este modelo.

Com o tempo, o ideal é que você desenvolva um arquivo próprio, com as opções que você usa mais frequentemente e comentários que lhe ajudem a lembrar como e em quais situações usar cada uma. Lembre-se de que, salvo eventuais diferenças entre as versões instaladas, um arquivo de configuração usado no Fedora ou Mandriva vai funcionar perfeitamente no Debian, Slackware, ou em qualquer outra distribuição que siga um nível mínimo de padrões. O software em si, o Postfix, será o mesmo, independentemente da distribuição usada.

Instalando um webmail

O Squirrelmail é um script de webmail escrito em php, que permite acessar as mensagens de um servidor imap via web. Ele é bem leve, tanto do ponto de vista dos recursos utilizados no servidor, quanto do ponto de vista dos clientes. As páginas geradas pelo webmail são simples páginas html, sem javascript nem nenhum outro recurso especial.

Isso o torna um campeão de compatibilidade, principalmente com os navegadores usados em PDAs e browsers antigos.

Para instalar o Squirrelmail você vai precisar do seguinte:

1- Um servidor Postfix (ou outro MTA suportado), com suporte a IMAP, o que inclui basicamente os pacotes "postfix" e "courier-imap". Siga as instruções anteriores para instalar o servidor de e-mails e criar as contas de usuários.

2- Um servidor Apache, com suporte a PHP4 instalado. Tanto faz usar o Apache 1.3 ou o Apache 2, o Squirrelmail roda em ambos, verifique apenas se o suporte a PHP está realmente instalado e funcionando.

Satisfeitos esses dois pré-requisitos, o Squirrelmail em si é bem simples de instalar. Você tem duas opções. Pode instalá-lo usando o gerenciador de pacotes da sua distribuição ou baixar o arquivo manualmente. A principal vantagem em usar o pacote incluído na distribuição é que a instalação é feita com checagem de dependências, o que é uma segurança a mais contra eventuais barbeiragens na configuração do Postfix ou do Apache.

No Debian, por exemplo, você pode instalá-lo usando o apt-get:

```
# apt-get install squirrelmail
```

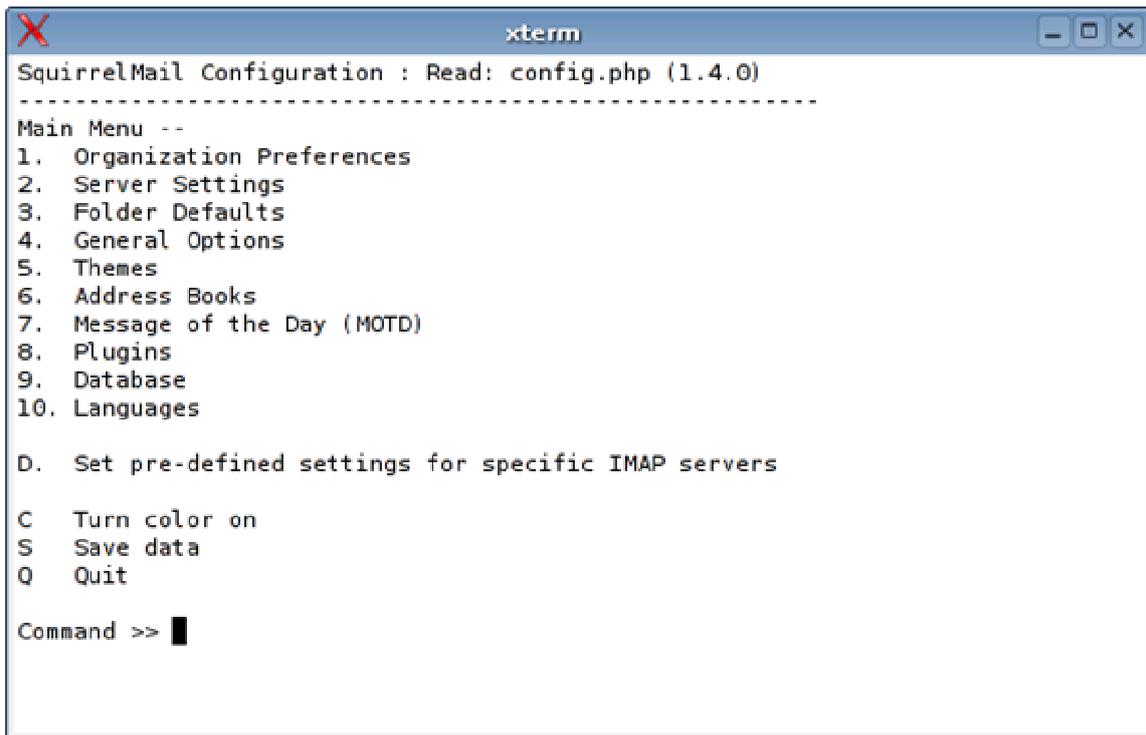
O Squirrelmail é instalado por padrão dentro da pasta `"/usr/share/squirrelmail"`, que fica fora da jurisdição do servidor web. Existem várias formas de fazer com que o webmail fique acessível apesar disso. Você pode, por exemplo, criar um link dentro da pasta `"/var/www/"` apontando para a pasta `"/usr/share/squirrelmail"`. Mas, uma forma mais elegante de ter o mesmo resultado, é adicionar as duas linhas abaixo no arquivo `"/etc/apache2/httpd.conf"`:

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

Aqui estamos criando uma pasta virtual "webmail/" no servidor web, que aponta para o arquivo `index.php` dentro da pasta real.

Ao baixar manualmente, pegue o arquivo no <http://www.squirrelmail.org/download.php> e copie o conteúdo do arquivo para uma pasta dentro do seu servidor web, como, por exemplo, `"/var/www/webmail"`; basta descompactar o arquivo, como no caso do phpBB. Não é necessário compilar nada. Opcionalmente, você pode usar a pasta `"/usr/share/squirrelmail"` e adicionar a entrada do alias no arquivo de configuração do Apache.

Depois de instalar, é preciso fazer a configuração básica do Squirrelmail, usando o utilitário `"squirrelmail-config"`. Se você instalou a partir do pacote, pode chamá-lo diretamente (como root) a partir do terminal. Se tiver instalado manualmente, execute o script `"configure"`, dentro da pasta do Squirrelmail.



```
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color on
S Save data
Q Quit

Command >> █
```

A configuração mínima inclui:

- a) Defina o nome da empresa, logotipo, URL, etc. na opção **1**.
- b) Defina o domínio do servidor (ex: minhaempresa.com) na opção **2**. Se você está configurando um servidor local, sem usar um domínio registrado, mantenha o default.
- c) Ainda na opção **2**, verifique se as configurações de servidor estão corretas. Elas devem ser:

- 3. Sendmail or SMTP : SMTP
- A. Update IMAP Settings : localhost:143 (other)
- B. Update SMTP Settings : localhost:25

O Squirrelmail pode ser configurado para utilizar outros servidores. Você pode usar o Sendmail (ou Qmail) no lugar do Postfix ou utilizar outros servidores IMAP além do Courier. As configurações acima são as que se aplicam no nosso caso, usando o postfix e o courier-imap.

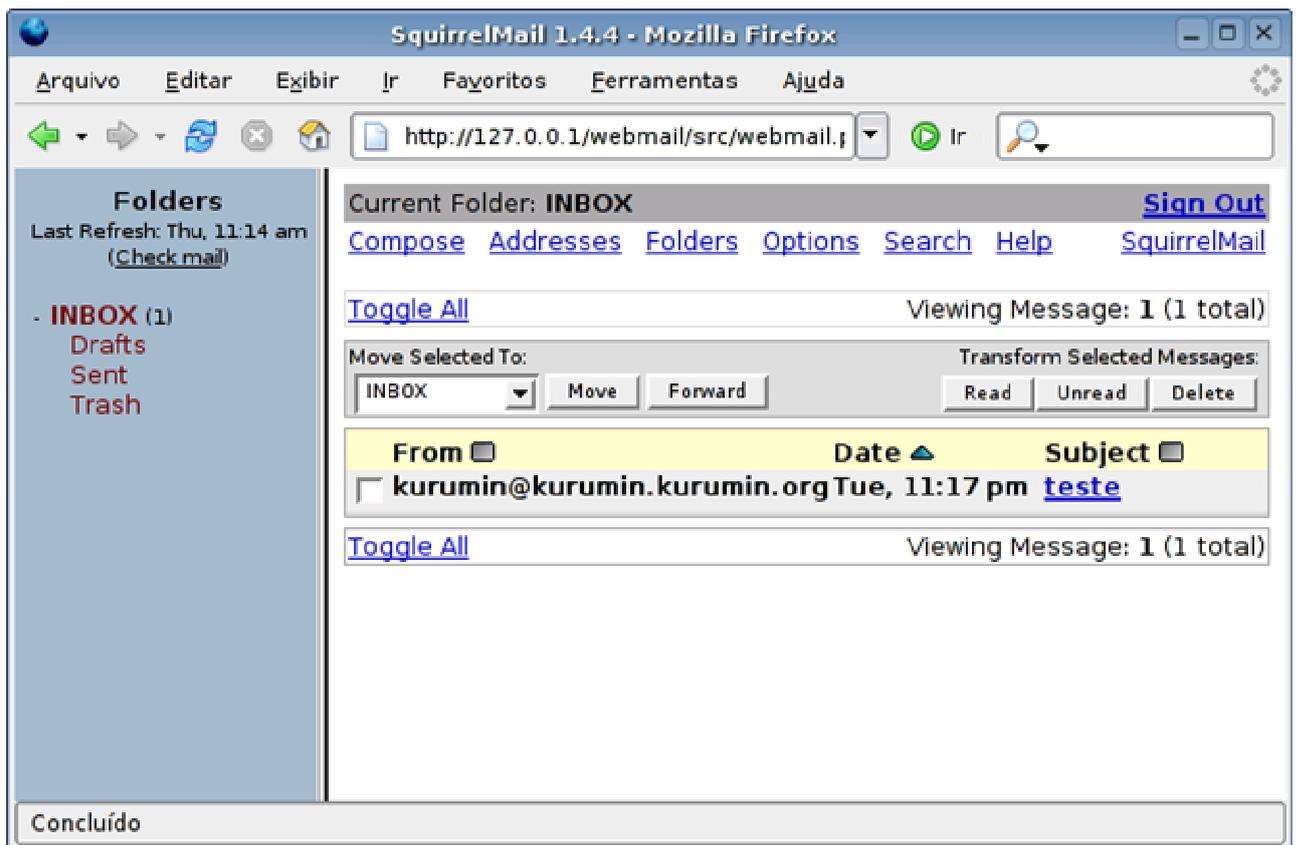
- d) Acesse a opção **D** no menu e indique qual servidor IMAP está utilizando. Lembre-se de que no exemplo estamos usando o Courier. Definir corretamente o servidor usado aqui permite que o Squirrelmail ative uma série de otimizações para cada servidor específico, que melhoram consideravelmente o desempenho. Ao terminar, use a opção **S** para salvar e **Q** para sair.

O Squirrelmail não é um serviço, ele é apenas uma aplicação que roda dentro do Apache. Depois de instalar os arquivos, acesse o endereço "http://127.0.0.1/webmail" e você verá a tela de login do Squirrelmail.



Se algo der errado neste ponto, verifique a instalação do Apache, se o suporte a PHP está realmente funcionando (lembre-se de que além de instalar o pacote, é necessário incluir a linha "LoadModule php4_module /usr/lib/apache/1.3/libphp4.so" ou similar no arquivo de configuração do Apache) e se a pasta de instalação, o link ou a entrada no arquivo de configuração para vincular a pasta webmail/ com a pasta onde estão os arquivos estão corretos.

Depois de logado, faça alguns testes, verificando se consegue mandar e-mails para contas em outros servidores e se consegue mandar e-mails de um usuário local para outro. Se o servidor já estiver disponível na internet, experimente enviar um e-mail (a partir de outra conta) para ele, usando inicialmente o endereço IP (ex: joao@200.220.123.32) e depois o nome de domínio registrado (joao@minhaempresa.com).



Caso a tela de login funcione, mas você tenha problemas ao logar, verifique as permissões de acesso da pasta de instalação do Squirrelmail, veja se ela não está com permissão de leitura apenas para o root. Lembre-se de que na maioria das distribuições o Apache roda sob o usuário "apache" ou "daemon" e não sob o usuário root, o que é inseguro.

Verifique também a configuração do Squirrelmail, veja se o serviço "**courier-imap**" está realmente inicializado. Observe que no Squirrelmail o login de acesso é apenas o nome do usuário (ex: joao) e não o endereço de e-mail completo. Ao configurar um servidor simples, onde as contas de acesso do sistema são usadas no servidor de e-mail, as senhas também são as mesmas.

Em algumas distribuições, depois de instalar o pacote courier-imap, é necessário rodar o comando "pw2userdb" para que as contas de usuários do sistema sejam corretamente incluídas como logins de acesso no servidor de e-mails. Verifique isso e reinicie o courier-imap novamente.

Uma última pegadinha é que, para que o servidor IMAP funcione, é necessário que exista um diretório chamado "Maildir" dentro do home de cada usuário, onde são armazenadas as mensagens. Este diretório contém uma estrutura própria e é criado usando o comando "maildirmake". Normalmente ele é criado automaticamente ao instalar os pacotes usados nas distribuições. Mas, em algumas, este procedimento precisa ser feito manualmente. É mais uma coisa que pode dar errado.

Se isso for necessário no seu caso, comece criando o diretório para o seu próprio usuário, ou o que for usar para testar o webmail:

```
$ maildirmake ~/Maildir
```

Execute agora o comando que cria a pasta dentro do diretório `/etc/skel`, de forma que os diretórios home de todos os novos usuários criados daqui em diante já sejam criados com ele:

Autenticando os clientes e Ativando o TLS

Autenticando os clientes

Originalmente, o Postfix determina os clientes que estão autorizados a enviar e-mails através do seu servidor de acordo com a configuração da linha "mynetworks", dentro do arquivo `main.cf`. Usando a linha "mynetworks = 127.0.0.0/8" ou "mynetworks = 127.0.0.1" o Postfix aceita apenas e-mails enviados a partir do próprio servidor, uma configuração ideal se os usuários enviam os e-mails através de um webmail instalado no próprio servidor, sem SMTP externo.

Você pode também permitir o envio a partir de qualquer micro da rede local, usando algo como "mynetworks = 192.168.0.0/24". O problema surge quando você precisa permitir o envio de e-mails para usuários espalhados pela web, conectados via ADSL, modem ou outras modalidades de conexão com IP dinâmico.

Imagine, por exemplo, o caso de um provedor de acesso que precisa permitir que seus usuários enviem e-mails usando seu SMTP, mesmo quando eles estiverem acessando através de outro provedor.

Você não pode simplesmente permitir o envio a partir de qualquer endereço, caso contrário seu servidor vai ser rapidamente descoberto pelos spammers, que começarão a utilizar toda a sua banda para enviar suas tentadoras ofertas de produtos. Pior, depois de algum tempo, seu servidor vai acabar caindo nas listas negras de endereços usados para envio de spam, fazendo com que seus próprios e-mails válidos passem a ser recusados por outros servidores.

A solução, nesse caso, é passar a autenticar os usuários, como faz a maioria dos provedores. Usamos então o SASL, que no Debian (Etch ou Sid) pode ser instalado via `apt-get`:

```
# apt-get install libsasl2 sasl2-bin libsasl2-modules libdb3-util procmail
```

Depois de instalar os pacotes, abra o arquivo `"/etc/default/saslauthd"`, onde vão as opções de inicialização do autenticador. O primeiro passo é substituir a linha "START=no" por:

```
START=yes
```

Adicione (ou modifique) também a linha:

```
MECHANISMS="pam"
```

Isso faz com que ele seja inicializado durante o boot e aceite a autenticação dos usuários.

Continuando, crie (ou edite) o arquivo `"/etc/postfix/sasl/smtpd.conf"` de forma que ele contenha apenas as linhas:

```
pwcheck_method: saslauthd  
mech_list: plain login
```

O pacote do Postfix usado no Debian Etch e no Ubuntu 6.10 (ou mais recente) e em outras distribuições derivadas deles, roda dentro de um chroot (ou jaula), o que melhora bastante a segurança, impedindo que qualquer eventual problema de segurança fique restrito ao servidor de e-mails, sem afetar o resto do sistema. Você notará que dentro da pasta `"/var/spool/postfix"` estão não apenas os diretórios com as filas de mensagens, mas também binários e bibliotecas de que o postfix precisa para funcionar.

O problema é que de dentro do seu chroot, o Postfix não tem acesso ao saslauthd, fazendo com que a autenticação dos usuários não funcione. O próprio saslauthd é necessário por que o Postfix (mesmo ao rodar fora do chroot) não tem acesso aos arquivos de senha do sistema e por isso não é capaz de autenticar os usuários por si só.

Para resolver este problema, precisamos criar a pasta `"/var/spool/postfix/var/run/saslauthd"`, utilizada pelo Postfix dentro do chroot e configurar o sasl para utilizá-la no lugar da pasta padrão. Desta forma, o Postfix consegue se comunicar com ele.

Este tipo de precaução de segurança parece algo complicado e desnecessário à primeira vista, mas é justamente por causa de truques como este que os servidores Linux acabam sendo tão seguros. Para começo de conversa, o Postfix é por si só bastante seguro. Mas, como os servidores de e-mail são um ponto comum de ataque, ele fica isolado dentro do chroot de forma que, mesmo na remota possibilidade de um cracker conseguir obter controle sobre o Postfix através um exploit remoto, ele não poderia fazer muita coisa. Para completar, o Postfix roda dentro de privilégios muito limitados, de forma que, mesmo que o cracker tenha muita sorte e a improvável falha de segurança no Postfix seja combinada com uma falha no sistema que o permita escapar do chroot, ele ainda assim não conseguiria fazer muita coisa. ;)

Comece criando o diretório:

```
# mkdir -p /var/spool/postfix/var/run/saslauthd
```

Abra agora o arquivo `"/etc/default/saslauthd"` (o mesmo onde substituímos o `"START=no"` por `"START=yes"`) e substitua a linha

```
OPTIONS="-c"
```

por:

```
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"
```

Reinicie o serviço para que as alterações entrem em vigor:

```
# /etc/init.d/saslauthd restart
```

Isso faz com que o SASL passe a utilizar o diretório dentro do chroot e o Postfix tenha acesso ao saslauthd e possa assim autenticar os usuários através dele. Note que o `"/var/spool/postfix"` é o diretório onde está o chroot. Esta é a localização padrão no Debian; ao usar outra distribuição, verifique se não está sendo usado outro diretório.

Só para garantir, adicione o postfix ao grupo sasl:

```
# adduser postfix sasl
```

Isso completa a configuração do SASL.

O passo seguinte é a configuração do Postfix. Abra o arquivo `"/etc/postfix/main.cf"` e adicione as linhas abaixo no final do arquivo. Ao reciclar um arquivo de configuração anterior, verifique se esta configuração já não foi incluída em outro ponto do arquivo:

```
smtpd_sasl_local_domain =  
smtpd_sasl_auth_enable = yes  
smtpd_sasl_security_options = noanonymous  
broken_sasl_auth_clients = yes  
smtpd_recipient_restrictions = permit_sasl_authenticated,  
permit_mynetworks,  
reject_unauth_destination  
smtpd_tls_auth_only = no
```

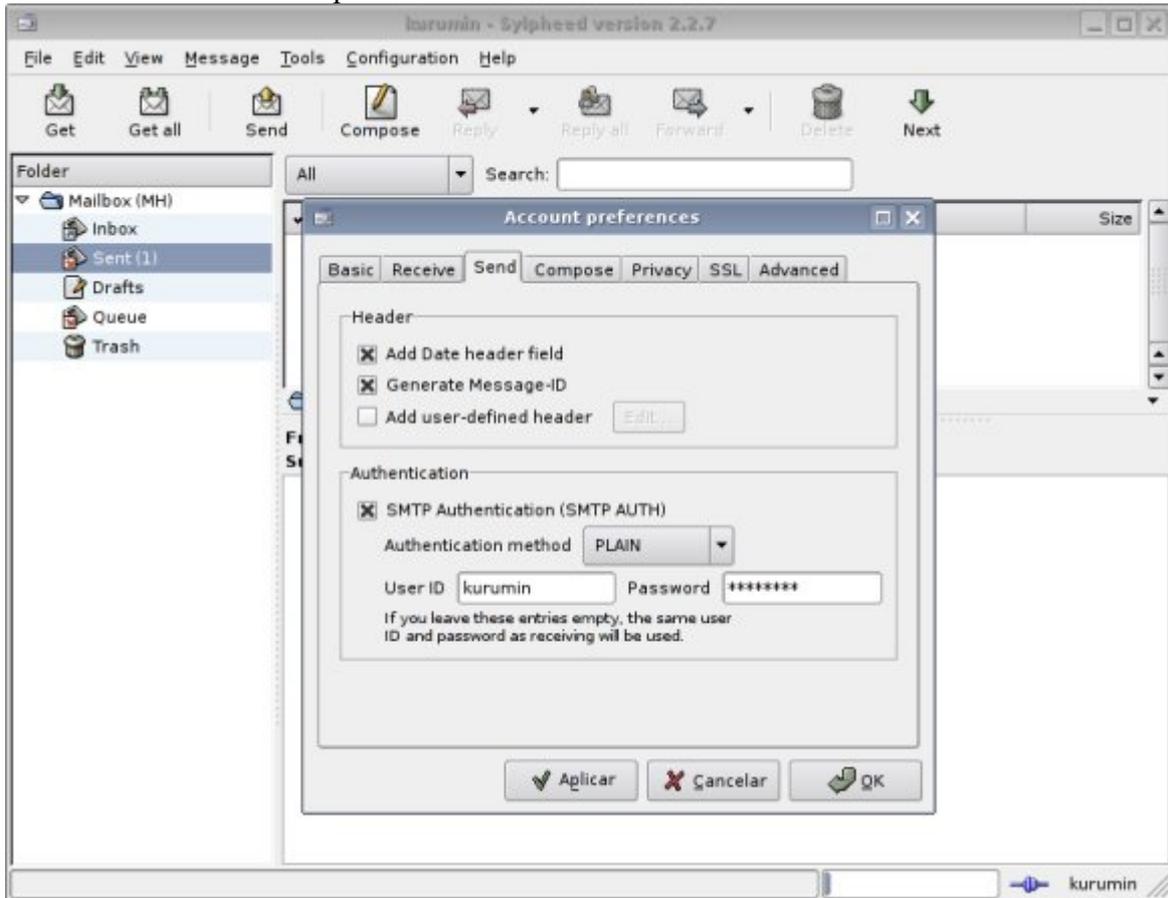
Feito isso, reinicie o Postfix para que as alterações entrem em vigor:

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

Por enquanto, o servidor suporta apenas autenticação em texto puro, sem encriptação. Este é o sistema "clássico", ainda usado por muito provedores de acesso, mas que possui problemas óbvios de segurança, já que alguém que consiga sniffar a rede local, poderia capturar as senhas dos usuários no momento em que eles tentassem baixar os e-mails.

Para testar, configure um cliente de e-mails qualquer para utilizar o endereço IP do servidor como SMTP (aqui estou usando o Sylpeed) e, nas configurações, ative a opção de autenticação para o servidor SMTP e escolha a opção "PLAIN" (login em texto puro).

Envie um e-mail de teste para confirmar se tudo está funcionando.



Ativando o TLS

O TLS (Transport Layer Security) adiciona segurança ao nosso sistema de autenticação, permitindo que os usuários possam baixar os e-mails sem medo, mesmo ao acessar a partir de redes públicas.

Em algumas distribuições (como no Debian Sarge), você precisa instalar o pacote "postfix-tls". Nas demais (incluindo o Debian Etch, que é a versão atual), ele já vem integrado ao pacote principal do Postfix.

O TLS trabalha utilizando um conjunto de chaves de encriptação e certificados, usados para criar o túnel encriptado e garantir a segurança da seção. O primeiro passo é criar este conjunto de arquivos.

Acesse o diretório "/etc/postfix/ssl" (crie-o se não existir) e rode os comandos abaixo, um de cada vez e nesta ordem. Durante a geração das chaves, será solicitado que você informe uma passphrase, uma senha que pode conter entre 4 e 8191 caracteres. Administradores paranóicos costumam usar passphrases bem grandes, mas não exagere, pois você precisará confirmá-la algumas vezes durante o processo. Os comandos são:

```
# mkdir /etc/postfix/ssl
# cd /etc/postfix/ssl/
# openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
# chmod 600 smtpd.key
```

```
# openssl req -new -key smtpd.key -out smtpd.csr
# openssl x509 -req -days 730 -in smtpd.csr -signkey smtpd.key -out smtpd.crt
# openssl rsa -in smtpd.key -out smtpd.key.unencrypted
# mv -f smtpd.key.unencrypted smtpd.key
# openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days
730
```

O "730" usado nas linhas determina a validade dos certificados, em dias. No caso, estou criando certificados válidos por dois anos. Depois deste prazo, os clientes começarão a receber um aviso ao se autenticarem, avisando que o certificado expirou e precisarei repetir o processo para atualizá-los. Se preferir, você pode usar um número mais alto, para gerar certificados válidos por mais tempo. Para gerar certificados válidos por 10 anos, por exemplo, substitua o "730" por "3650".

Continuando, abra novamente o arquivo `"/etc/postfix/main.cf"` e adicione as linhas abaixo (sem mexer nas linhas referentes ao SASL que adicionamos anteriormente):

```
smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
tls_random_source = dev:/dev/urandom
```

Reinicie o Postfix para que as alterações entrem em vigor:

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

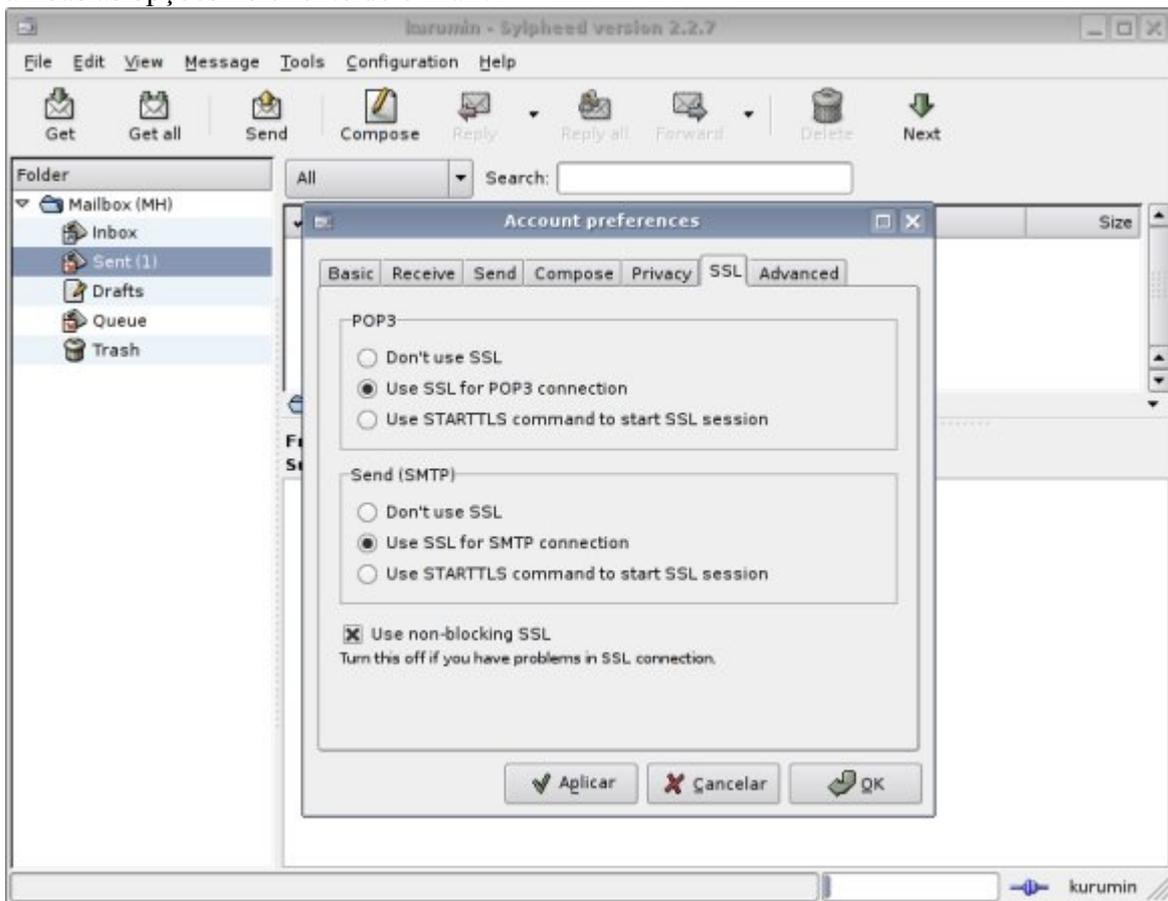
Para que os clientes consigam se autenticar no servidor, é necessário instalar o pacote "courier-authdaemon" e o "courier-ssl", além dos pacotes courier-pop, courier-pop-ssl, courier-imap, courier-imap-ssl que vimos anteriormente. Você pode usar o comando abaixo para instalar de uma vez todos os pacotes necessários:

```
# apt-get install courier-authdaemon courier-base courier-imap courier-imap-ssl \
courier-pop courier-pop-ssl courier-ssl gamin libgamin0 libglib2.0-0
```

Para testar, ative o uso do SSL para o servidor SMTP dentro das preferências do seu cliente de e-mail. No caso do Thunderbird, por exemplo, marque a opção "Usar Conexão Segura > TLS" dentro do menu "Enviar", nas configurações da conta. O cliente de e-mail exibirá alguns avisos sobre a validade do certificado, o que é normal, já que estamos utilizando um certificado "self-signed", ou seja, um certificado "caseiro", que não é reconhecido por nenhuma autoridade certificadora. Empresas como a Verisign vendem certificados reconhecidos, mas os preços são proibitivos fora de grandes instalações.

Com o TLS, A autenticação continua funcionando da mesma forma, mas agora todos os dados são transmitidos de forma segura. Lembre-se de que ao instalar o courier, já

ativamos também o suporte a SSL para o IMAP e POP3, de forma que você pode ativar ambas as opções no cliente de e-mail:



Aqui está um exemplo de arquivo `/etc/postfix/main.cf` completo, incluindo a configuração do SASL e do TLS:

```
# /etc/postfix/main.cf
```

```
myhostname = etch.kurumin.com.br
mydomain = kurumin.com.br
append_dot_mydomain = no
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = etch.kurumin.com.br, kurumin.com.br, localhost
relayhost =
mynetworks = 127.0.0.0/8
home_mailbox = Maildir/
mailbox_command =
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
message_size_limit = 20000000
mailbox_size_limit = 0

smtpd_sasl_local_domain =
```

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_sasl_authenticated,
permit_mynetworks,
reject_unauth_destination
smtpd_tls_auth_only = no
```

```
smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
tls_random_source = dev:/dev/urandom
```

O arquivo **/etc/default/saslauthd** (depois de removidos os comentários), ficaria:

```
START=yes
MECHANISMS="pam"
MECH_OPTIONS=""
THREADS=5
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"
```

O arquivo **/etc/postfix/sasl/smtpd.conf**, que vimos no início, continua com apenas as duas linhas:

```
pwcheck_method: saslauthd
mech_list: plain login
```

Adicionando um antivírus

Depois que o servidor de e-mails estiver funcionando, é interessante instalar um antivírus para proteger as estações Windows. Este é um detalhe interessante: existem vários bons antivírus para Linux, mas todos são destinados a justamente encontrar vírus for Windows em compartilhamentos de rede, páginas web e arquivos, e-mails, etc. Até hoje, o Linux (tanto como servidor, quanto desktop) tem se mantido como uma plataforma livre de vírus dignos de nota, daí a ausência de soluções neste sentido. A demanda simplesmente não existe.

Uma das melhores opções é o **Clamav**, que possui uma lista de definições atualizada com uma frequência muito grande e oferece um recurso de atualização automática. O Clamav escaneia as mensagens que passam pelo servidor, removendo as mensagens com arquivos infectados. Ele serve tanto para proteger clientes Windows da rede, quanto para reduzir o tráfego de mensagens inúteis.

Para instalar o antivírus, basta instalar o pacote "**clamav**". Ele não costuma mudar de nome entre as distribuições. No Debian, você precisa instalar também o pacote "**clamav-daemon**", em outras distribuições este componente faz parte do pacote principal.

Para utilizar o Clamav em conjunto com o Postfix, de forma que todos os e-mails passem primeiro pelo antivírus, e só depois sejam encaminhados para as caixas postais dos usuários, é necessário instalar também o pacote "**amavisd-new**".

O Amavisd "intercepta" as novas mensagens, entregando-as ao executável do Clamav. De acordo com a configuração, as mensagens com arquivos infectados podem ser simplesmente deletadas, ou colocadas em uma pasta de quarentena. Lembre-se de que quase todas as mensagens com arquivos infectados são enviadas automaticamente pelos vírus da moda, como uma forma de se espalharem, por isso não existe muito sentido em preservá-las. O Amavisd é um software complicado de instalar manualmente, é necessário alterar vários scripts e arquivos de inicialização e configurar corretamente as permissões de várias pastas. Além de trabalhoso, o processo é muito sujeito a erros, por isso é sempre recomendável utilizar os pacotes incluídos nas distribuições, onde o trabalho já está feito.

A comunicação entre o Amavisd e o Clamav é feita automaticamente, mas é necessário configurar o Postfix para direcionar os novos e-mails para o Amavisd, para que o trio comece a trabalhar em conjunto. Para isso, é necessário adicionar as linhas abaixo no final do arquivo "**/etc/postfix/master.cf**" (note que este arquivo é diferente do main.cf que configuramos anteriormente). Estas linhas estão incluídas no arquivo "**/usr/share/doc/amavisd-new/README.postfix**"; você pode copiá-las a partir do arquivo, ao invés de escrever tudo:

```
smtp-amavis unix - - n - 2 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-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 smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks=127.0.0.0/8
-o strict_rfc821_envelopes=yes
-o smtpd_error_sleep_time=0
-o smtpd_soft_error_limit=1001
-o smtpd_hard_error_limit=1000
```

Adicione também a linha abaixo ao "**/etc/postfix/main.cf**":

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

É preciso verificar também os arquivos `"/etc/clamav/clamd.conf"` e `"/etc/amavis/amavisd.conf"`. Em muitas distribuições eles são configurados corretamente ao instalar os pacotes, mas em outras é preciso fazer as alterações manualmente.

No arquivo `"/etc/clamav/clamd.conf"`, verifique se a linha abaixo está presente e descomentada:

```
LocalSocket /var/run/clamav/clamd.ctl
```

No arquivo `"/etc/amavis/amavisd.conf"`, verifique se as linhas abaixo estão descomentadas. Este arquivo inclui vários exemplos que permitem usar diferentes antivírus, por isso é um pouco extenso. Use a função de pesquisar do editor de textos para ir direto ao ponto:

```
### http://www.clamav.net/  
['Clam Antivirus-clamd',  
\&ask_daemon, ["CONTSCAN {} \n", "/var/run/clamav/clamd.ctl"],  
qr/\bOK$/, qr/\bFOUND$/,  
qr/^.*?: (?!\bInfected Archive)(.*) FOUND$/ ],
```

O último passo é fazer com que o Amavis tenha acesso aos arquivos de log e diretórios de trabalho do Clamav. No Debian e em muitas outras distribuições, basta adicionar o usuário **"clamav"** no grupo **"amavis"**, a solução mais rápida e limpa:

```
# adduser clamav amavis
```

Caso você esteja usando outra distribuição, onde essa primeira alteração não funcione, resta fazer do jeito sujo, alterando manualmente as permissões de acesso dos diretórios:

```
# chown -R amavis:amavis /var/clamav  
# chown -R amavis:amavis /var/log/clamav  
# chown -R amavis:clamav /var/run/clamav
```

Depois de terminar a configuração, reinicie todos os serviços, de forma que tudo entre em vigor:

```
# /etc/init.d/postfix restart  
# /etc/init.d/clamav-daemon restart  
("/etc/init.d/clamd restart" no Fedora e Mandriva)  
# /etc/init.d/clamav-freshclam restart  
("/etc/init.d/freshclam restart" no Fedora e Mandriva)  
# /etc/init.d/amavis restart
```

Experimente tentar enviar agora um e-mail contendo um arquivo infectado qualquer para uma conta qualquer do seu servidor. O e-mail simplesmente não vai chegar ao destino. O arquivo `"/usr/share/doc/amavisd-new/README.postfix"` contém uma string de texto que dispara o antivírus e pode ser usada para simular um e-mail infectado.

Checando o conteúdo do arquivo `"/var/log/clamav/clamav.log"`, você verá uma entrada indicando que um e-mail infectado foi encontrado:

Thu Ago 26 19:55:49 2005 -> /var/lib/amavis/amavis-20050526T195549-07338/parts/part-00001: Eicar-Test-Signature FOUND

O próximo passo é instalar o Spamassassin que funciona como um filtro antispam automático, que utiliza uma blacklist com endereços IP e conteúdos de mensagem catalogados. Esta lista funciona de forma semelhante à de um programa antivírus, é atualizada pela equipe de desenvolvimento e atualizada de forma automática. Para instalar:

```
# apt-get install spamassassin
```

No Debian, por padrão, ele fica inativo depois de instalado, talvez como uma precaução para evitar consumir muitos recursos em micros onde ele é instalado acidentalmente junto com outros servidores. Para ativá-lo, edite o arquivo `"/etc/default/spamassassin"`, mudando a opção `"ENABLED=0"` para `"ENABLED=1"`:

```
# Change to one to enable spamd  
ENABLED=1
```

Para finalizar, inicie o serviço `"spamassassin"` (ou `"spamd"` em muitas distribuições):

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

Falta agora configurar o Amavis para utilizar também o Spamassassin ao receber novas mensagens. Agora os e-mails passarão pelos dois filtros, seqüencialmente.

Abra novamente o arquivo: `"/etc/amavis/amavisd.conf"`. Na seção 1, por volta da linha 160, **comente** (#) a linha:

```
@bypass_spam_checks_acl = qw( . );
```

Essa linha desativa a checagem de spam. Ela vem descomentada por padrão, pois nem todo mundo utiliza o Amavis em conjunto com o Spamassassin. Ao comentá-la, a checagem é ativada.

Na seção 4, por volta da linha 400, procure pela linha: `"$final_spam_destiny ="`.

Essa linha configura o que será feito com as mensagens marcadas como spam. Ela tem três valores possíveis:

D_PASS – Entrega a mensagem normalmente, incluindo apenas a palavra "SPAM" no subject. Isso permite que os próprios usuários configurem o filtro local do leitor de e-mails para remover as mensagens caso estejam incomodando. Nenhum filtro antispam é perfeito, sempre algumas mensagens legítimas acabam sendo marcadas como spam. Esta opção minimiza o problema, deixando a remoção das mensagens por conta dos usuários.

D_DISCARD – Descarta a mensagem. Esta é a opção mais usada, mas ao mesmo tempo a mais perigosa, pois mensagens "boas" marcadas como spam vão simplesmente sumir, sem deixar nenhum aviso ao remetente ou destinatário.

D_REJECT – Esta terceira opção também descarta a mensagem, mas envia uma notificação ao emissor. Isso permite que o emissor de uma mensagem "boa",

acidentalmente classificada como spam, receba um aviso de que ela foi descartada e tenha a oportunidade de reenviá-la novamente de outra forma. Mas, por outro lado, como a maioria dos spams são enviados a partir de endereços falsos, isso acaba sendo um desperdício de banda.

Ao usar a opção que descarta as mensagens, a opção fica:

```
$final_spam_destiny = D_DISCARD
```

Mais adiante, na seção 7, por volta da linha 1100, você encontrará mais opções relacionadas ao Spamassassin.

Finalmente, é preciso transferir o ownership dos arquivos do Spamassassin para o Amavis (assim como no caso do Clamav), para que ele possa executar corretamente suas funções:

```
# chown -R amavis:amavis /usr/share/spamassassin
```

Não se esqueça de reiniciar os serviços com o comando:

```
# /etc/init.d/spamassassin restart  
("/etc/init.d/spamd restart" em muitas distribuições)  
# /etc/init.d/amavis restart
```

Configurando o DNS Reverso

O DNS reverso é um recurso que permite que outros servidores verifiquem a autenticidade do seu servidor, checando se o endereço IP atual bate com o endereço IP informado pelo servidor DNS. Isso evita que alguém utilize um domínio que não lhe pertence para enviar spam, por exemplo.

Qualquer um pode enviar e-mails colocando no campo do remetente o servidor do seu domínio, mas um servidor configurado para checar o DNS reverso vai descobrir a farsa e classificar os e-mails forjados como spam.

O problema é que os mesmos servidores vão recusar seus e-mails, ou classificá-los como spam, caso você não configure seu servidor DNS corretamente para responder às checagens de DNS reverso. Chegamos a um ponto em que o problema do spam é tão severo que quase todos os servidores importantes fazem esta checagem, fazendo com que, sem a configuração, literalmente metade dos seus e-mails não seja entregue.

O primeiro passo é checar os arquivos **"/etc/hostname"** e **"/etc/hosts"** (no servidor), que devem conter o nome da máquina e o domínio registrado.

O arquivo **"/etc/hostname"** deve conter apenas o nome da máquina, como em:

```
servidor
```

No Fedora e em algumas outras distribuições, o nome da máquina vai dentro do arquivo **"/etc/sysconfig/network"**.

O arquivo "/etc/hosts" deve conter duas entradas, uma para a interface de loopback, o 127.0.0.1 e outra para o IP de internet do seu servidor, que está vinculado ao domínio, como em:

```
127.0.0.1 localhost.localdomain localhost
64.234.23.12 etch.kurumin.com.br etch
```

A partir daí, falta adicionar a zona reversa no bind complementando a configuração do domínio, que já fizemos. Começamos adicionando a entrada no "/etc/bind/named.conf" ou "/etc/bind/named.conf.local":

```
zone "23.234.64.in-addr.arpa" {
type master;
file "/etc/bind/db.kurumin.rev";
};
```

No nosso exemplo, o endereço IP do servidor é 64.234.23.12. Se retiramos o último octeto e escrevemos o restante do endereço de trás pra frente, temos justamente o "23.234.64" que usamos no registro reverso. A terceira linha indica o arquivo em que a configuração do domínio reverso será salva. Nesse caso, indiquei o arquivo "db.kurumin.rev", mas você pode usar qualquer nome de arquivo.

Este arquivo "db.kurumin.rev" é bem similar ao arquivo com a configuração do domínio. As três linhas iniciais são idênticas (incluindo o número de sincronismo), mas ao invés de usar o "A" para relacionar o domínio e cada subdomínio ao IP correspondente, usamos a diretiva "PTR" para relacionar o endereço IP de cada servidor ao domínio (é justamente por isso que chamamos de DNS reverso ;).

No primeiro arquivo, usamos apenas os três primeiros octetos do endereço (a parte referente à rede), removendo o octeto final (o endereço do servidor dentro da rede). Agora, usamos apenas o número omitido da primeira vez.

O IP do servidor é "64.234.23.12"; removendo os três primeiros octetos, ficamos apenas com o "12". Temos também o endereço do DNS secundário, que é 64.234.23.13, de onde usamos apenas o "13". Relacionando os dois a seus respectivos domínios, o arquivo fica:

```
@ IN SOA etch.kurumin.com.br. hostmaster.kurumin.com.br. (
2006040645 3H 15M 1W 1D )
NS etch.kurumin.com.br.
NS ns1.kurumin.com.br.
12 PTR kurumin.com.br.
13 PTR ns1.kurumin.com.br.
```

Caso você não esteja usando um DNS secundário, é só omitir as linhas referentes a ele, como em:

```
@ IN SOA etch.kurumin.com.br. hostmaster.kurumin.com.br. (
2006040645 3H 15M 1W 1D )
NS etch.kurumin.com.br.
12 PTR kurumin.com.br.
```

Depois de terminar, reinicie o Bind e verifique usando o **dig**. Comece checando o domínio, como em:

```
# dig kurumin.com.br
```

Na resposta, procure pela seção "ANSWER SECTION", que deverá conter o IP do servidor, como configurado no bind:

```
:: ANSWER SECTION:  
kurumin.com.br. 86400 IN A 64.234.23.12
```

Faça agora uma busca reversa pelo endereço IP, adicionando o parâmetro "-x", como em:

```
# dig -x 64.234.23.12
```

Na resposta você verá:

```
:: ANSWER SECTION:  
12.23.234.64.in-addr.arpa. 86400 IN PTR kurumin.com.br.
```

Ou seja, com o DNS reverso funcionando, o domínio aponta para o IP do servidor e o IP aponta para o domínio, permitindo que os outros servidores verifiquem a autenticidade do seu na hora de receber e-mails provenientes do seu domínio.

Lembre-se que seus e-mails podem ser classificados como spam também se seu IP estiver marcado em alguma blacklist. Você pode verificar isso rapidamente no <http://rbls.org/>.

Você vai notar, por exemplo, que praticamente qualquer endereço IP de uma conexão via ADSL ou modem vai estar listado, muitas vezes "preventivamente", já que é muito comum que conexões domésticas sejam usadas para enviar spam. É recomendável verificar periodicamente os IP's usados pelo seu servidor, além de verificar qualquer novo IP ou link antes de contratar o serviço.

Mais uma consideração importante sobre a configuração do DNS reverso é que você precisa ter autoridade sobre a faixa de IP's para que a configuração funcione.

Quando você aluga um servidor dedicado, é de praxe que receba uma pequena faixa de endereços IP, configurada usando uma máscara complexa, como a "**255.255.255.248**", onde você fica com uma faixa de 5 IP's diferentes (na verdade são 8, mas destes apenas 6 são válidos e um é usado como default gateway, para que seu servidor possa acessar a internet através da rede do datacenter) como, por exemplo, do "72.213.43.106" até o "72.213.43.110".

Isso também é comum ao alugar um link dedicado, onde (dependendo do plano) você recebe uma faixa completa, com 254 IP's, ou uma faixa reduzida, com máscara "255.255.255.240" (14 IP's), "255.255.255.248" (6 IP's) ou "255.255.255.252" (2 IP's).

Em qualquer um dos casos, o fornecedor deve delegar a autoridade sobre a sua faixa de endereços, para que a configuração que vimos aqui funcione. Sem isto, é o servidor deles

quem responde pela sua faixa, retornando um erro qualquer, sem que seu servidor DNS tenha chance de fazer seu trabalho.

Se você alugou um servidor ou um link dedicado e percebeu que a autoridade sobre a faixa não foi delegada, minha primeira sugestão é que você troque de fornecedor, pois essa é uma configuração básica. Se não fazem nem a delegação corretamente, é provável que o serviço deixe a desejar em outros aspectos. Se isso não for possível, entre em contato com eles cobrando a configuração.

Alguns provedores preferem configurar eles mesmos o DNS reverso para seu domínio. Nesse caso, vão apenas lhe pedir a configuração e ativar o reverso no DNS deles. Essa solução não é ideal, pois você vai precisar entrar em contato com o suporte cada vez que precisar fazer uma alteração, mas é melhor do que nada. Essa é também a única opção em planos em que você recebe um único IP fixo, ao invés de uma faixa de endereços.

No caso de planos de acesso doméstico, onde você recebe um único IP, quase sempre é impossível configurar o DNS reverso, pois você não tem autoridade sobre a faixa de IP's e a operadora não vai querer fazer a configuração para você sem que você pague mais um plano empresarial.

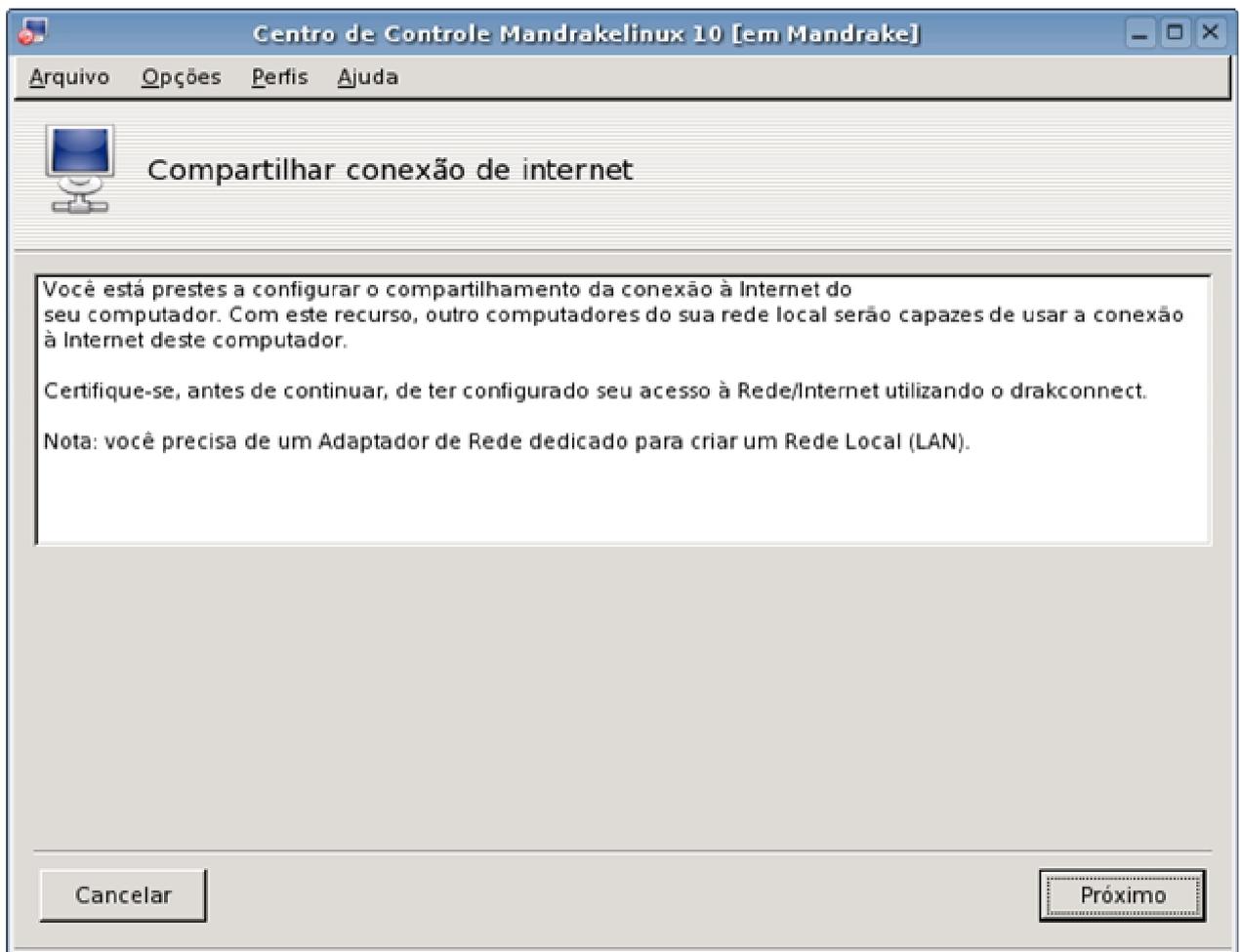
Configurando um servidor Proxy com o Squid

O Squid é um **servidor proxy**. Ele permite compartilhar a conexão entre vários micros, servindo como um intermediário entre eles e a Internet. Usar um proxy é diferente de simplesmente compartilhar a conexão diretamente, via NAT.

Ao compartilhar via NAT os micros da rede acessam a internet diretamente, sem restrições. O servidor apenas repassa as requisições recebidas, como um garoto de recados. O proxy é como um burocrata que não se limita a repassar as requisições: ele analisa todo o tráfego de dados, separando o que pode ou não pode passar e guardando informações para uso posterior.

Compartilhar a conexão via NAT é mais simples do que usar um proxy como o Squid sob vários aspectos. Você compartilha a conexão no servidor, configura os clientes para o utilizarem como gateway e pronto.

Por exemplo, para compartilhar a conexão via NAT no Mandrake você usaria o "Compartilhar conexão com a Internet" dentro do Painel de Controle e configuraria os clientes para usar o endereço IP do servidor (192.168.0.1 por exemplo) como gateway. No Kurumin você usaria o ícone mágico no Iniciar > Internet > Compartilhar Conexão e Firewall.



Ao usar um proxy, além da configuração da rede 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.

A configuração do proxy muda de navegador para navegador. No Firefox por exemplo você encontra em Editar > Preferências > Geral > Proxy. No IE a configuração está em Opções da Internet > Opções > Configurações da Lan > Usar um servidor Proxy.



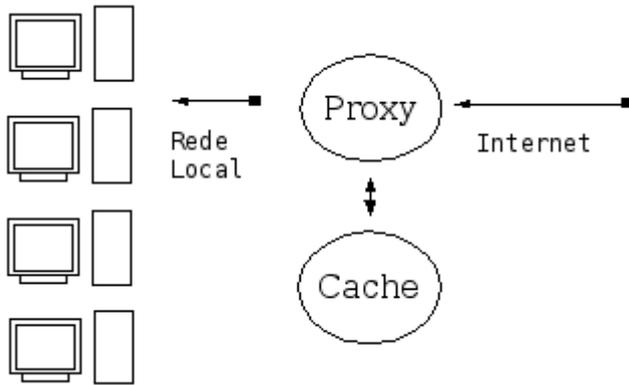
Além do navegador, outros programas, podem ser configurados para trabalhar através do proxy: Clientes de ICQ e MSN e até programas P2P.

As vantagens de usar um proxy são basicamente três:

1- É possível impor restrições de acesso com base no horário, login, endereço IP da máquina e outras informações e bloquear páginas com conteúdo indesejado.

2- 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 bastante banda na conexão com a Internet e tornando o acesso mais rápido sem precisar investir numa conexão mais rápida.

Hoje em dia os sites costumam usar páginas dinâmicas, onde o conteúdo muda a cada visita, mas mesmo nestes casos o proxy dá uma ajuda, pois embora o html seja diferente a cada visita, e realmente precise ser baixado de novo, muitos componentes da página, como ilustrações, banners e animações em flash podem ser aproveitadas do cache, diminuindo o tempo total de carregamento.



Dependendo da configuração, o proxy pode apenas 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. Ao invés de ter que baixar o Service Pack XYZ do Windows XP ou o OpenOffice nos 10 micros da rede, você vai precisar baixar apenas no primeiro, pois os outros 9 vão baixar a partir do cache do squid.

3- Uma terceira vantagem de usar um proxy é que ele loga todos os acessos. Você pode visualizar os acessos posteriormente usando o **Sarg**, assim você sabe quem acessou quais páginas e em que horários. Além de tudo, o Squid é dedo duro :-P

Squid User Access Reports
 Period: 2003May25-2003May26
 User: 192.168.0.3
 Sort: BYTES,
 User Report

date/time	ACCESSED SITE	CONNECT	BYTES	%BYTES	IN-CACHE-OUT	USED TIME	
	www.guiadohardware.net	17	127,994	96.90%	0.00% 100.00%	00:00:33	
	status.icq.com	8	3,506	2.65%	13.46% 86.54%	00:00:44	
	www.guiadohardware.info	2	586	0.44%	0.00% 100.00%	00:00:21	
TOTAL			27 132,086	18.25%	0.36%	99.64%	00:01:40
AVERAGE			78 361,996				00:03:22

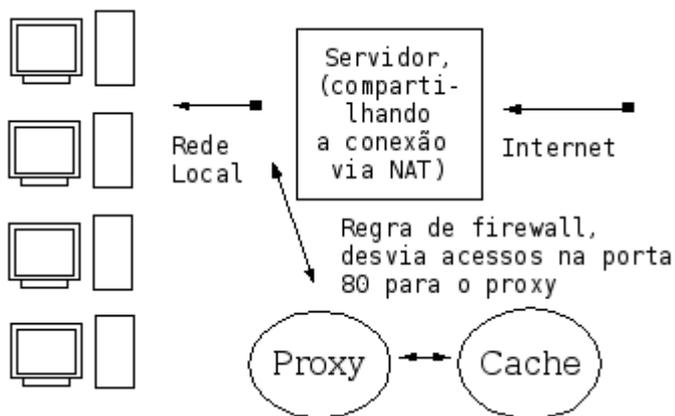
Generated by sarg-1.4.1 25Apr2003 on May/26/2003 02:05

Mesmo assim, você pode estar achando que as vantagens não vão compensar o trabalho de sair configurando micro por micro, programa por programa para usar o proxy e é mais

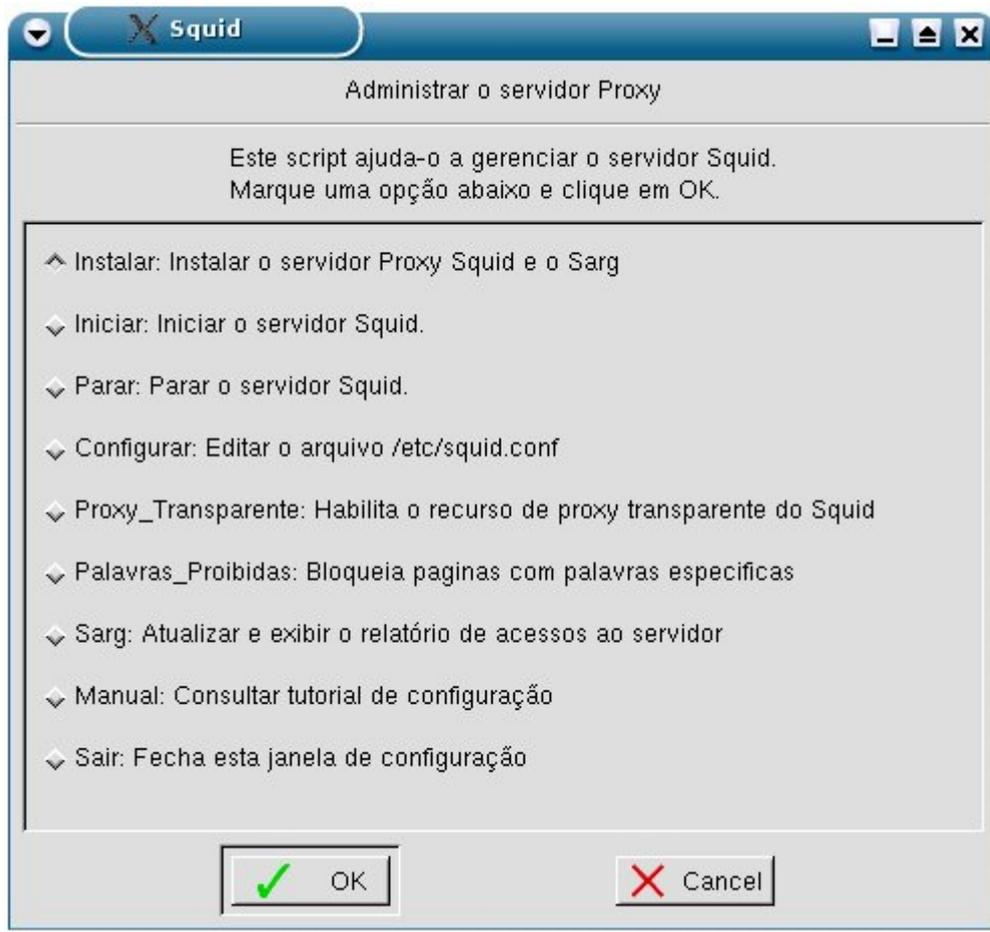
fácil simplesmente compartilhar via NAT. Mas, existe a possibilidade de juntar as vantagens das formas de compartilhamento, configurando um **proxy transparente** como veremos adiante.

Ao usar um proxy transparente, você tem basicamente uma conexão compartilhada via NAT, com a mesma configuração básica nos clientes. O proxy entra na história como uma espécie add-on.

Uma regra de firewall 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 (incluindo o negócio do cache dos arquivos do Windows update, downloads e do apt-get), sem que você precise fazer nenhuma configuração adicional nos clientes :-)



A algum tempo atrás, escrevi um script de configuração para o Kurumin, disponível no painel de configuração de servidores. Este script permite instalar o proxy, estabelecer algumas restrições de acesso e ativar o proxy transparente de forma simples:



Neste tutorial vou ensinar como fazer isso e criar regras mais elaboradas de restrição de acesso em qualquer distribuição.

Instalando o Squid

O Squid é composto de um único pacote, por isso a instalação é simples. Se estiver no Mandrake, instale com um:

```
# urpmi squid
```

Se estiver no Debian ou outra distribuição baseada nele, use o apt-get:

```
# apt-get install squid
```

Toda a configuração do Squid é feita num único arquivo, o **/etc/squid/squid.conf**.

Caso você esteja usando uma versão antiga do Squid, como a incluída no Debian Woody por exemplo, o arquivo pode ser o **/etc/squid.conf**. Apesar da mudança na localização do arquivo de configuração, as opções descritas aqui devem funcionar sem maiores problemas.

O arquivo original, instalado junto com o pacote é realmente enorme, contém comentários e exemplos para quase todas as opções disponíveis. Ele pode ser uma leitura interessante se você já tem uma boa familiaridade com o Squid, mas de início é melhor começar com um arquivo de configuração mais simples, apenas com as opções mais usadas.

Em geral cada distribuição inclui uma ferramenta diferente para a configuração do Proxy, como o ícone mágico que incluí no Kurumin.

Uma das mais usadas é o **Webmin**. A função destas ferramentas é disponibilizar as opções através de uma interface gráfica e gerar o arquivo de configuração com base nas opções escolhidas.

Em alguns casos estas ferramentas ajudam bastante, mas como elas mudam de distribuição para distribuição, é mais produtivo aprender a trabalhar direto no arquivo de configuração, que não é tão complicado assim.

Pra começar renomeie o arquivo padrão:

```
# mv /etc/squid/squid.conf /etc/squid/squid.conf.velho
```

e crie um novo arquivo **/etc/squid/squid.conf**, com apenas as quatro linhas abaixo:

```
-----  
    http_port 3128  
    visible_hostname kurumin  
    acl all src 0.0.0.0/0.0.0.0  
    http_access allow all  
-----
```

Estas linhas são o suficiente para que o Squid "funcione". Como você percebeu, aquele arquivo de configuração gigante tem mais uma função informativa, citando e explicando as centenas de opções disponíveis. As quatro linhas dizem o seguinte:

- **http_port 3128**: A porta onde o servidor vai ficar disponível
- **visible_hostname kurumin**: O nome do servidor
- **acl all src 0.0.0.0/0.0.0.0 e http_access allow all**: Estas duas linhas criam uma acl (uma política de acesso) chamada "all" (todos) incluindo todos os endereços IP possíveis e permite que qualquer um dentro desta lista use o proxy. Ou seja, ela permite que qualquer um use o proxy, sem limitações.

Para testar o Squid, habilite o servidor com o comando:

```
# service squid start
```

Se você estiver no **Debian**, use o comando:

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

Se estiver no **Slackware**, o comando será:

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

Configure um navegador, no próprio servidor para usar o proxy, através do endereço 127.0.0.1 (o localhost), porta 3128 e teste a conexão.

Se tudo estiver ok você conseguirá acessar o proxy também através dos outros micros da rede local, basta configurar os navegadores para usarem o proxy.

Criando uma configuração básica

O problema é que com apenas estas quatro linhas o proxy está muito aberto. Se você deixar o servidor proxy ativo no próprio servidor que compartilha a conexão e não tiver nenhum firewall ativo, alguém na Internet poderia usar o seu proxy, o que naturalmente não é desejado. O proxy deve ficar ativo apenas para a rede local. Vamos gerar então um arquivo mais completo, permitindo que apenas os micros da rede local possam usar o proxy e definindo mais algumas políticas de segurança.

Aqui eu já aproveitei algumas linhas do arquivo original, criando regras que permitem o acesso a apenas algumas portas, e não a qualquer coisa como antigamente:

```
http_port 3128
visible_hostname kurumin

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

Veja que agora criei duas novas acl's. A acl "localhost" contém o endereço 127.0.0.1, você usa para usar o proxy localmente, a partir do próprio servidor e a acl "rede local" que inclui os demais micros da rede local.

Você deve substituir o " **192.168.1.0/24**" pela a faixa de endereços IP e a máscara de sub-rede usada na sua rede local (o 24 equivale à máscara 255.255.255.0).

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
```

Lembra-se da acl "all", que contém todo mundo? Vamos usa-la para especificar que quem não se enquadrar nas duas regras acima (ou seja, micros não autorizados, da internet) não poderá usar o proxy:

http_access deny all

Esta linha deve ir no final do arquivo, depois das outras duas. O ordem é importante, pois o squid interpreta as regras na ordem em que são colocadas no arquivo. Se você permite que o micro X acesse o proxy, ele acessa, mesmo que uma regra mais abaixo diga que não.

Se você adicionasse algo como:

```
acl redelocal src 192.168.1.0/24
http_access allow redelocal
http_access deny redelocal
```

Os micros da rede local continuariam acessando, pois a regra que permite vem antes da que proíbe.

Configurando o cache de páginas e arquivos

Outra coisa importante é configurar o cache do proxy. O squid trabalha com dois tipos de cache:

- 1- Um cache rápido, feito usando parte da memória RAM do servidor
- 2- Um cache um pouco mais lento porém maior, feito no HD

O cache na memória RAM é ideal para armazenar arquivos pequenos, como páginas html e imagens, que serão entregues instantaneamente para os clientes. O cache no HD é usado para armazenar arquivos maiores, como downloads, arquivos do Windows update e pacotes baixados pelo apt-get.

O cache na memória RAM é sempre relativamente pequeno. Num servidor não dedicado ou seja, uma máquina que é usada para fazer outras coisas, mas roda também o proxy, você vai reservar coisa de 16 ou 32 MB de RAM para o cache, para evitar que o cache do squid coma toda a memória RAM deixando o micro lento.

Se você tiver uma rede muito grande e preferir deixar um micro dedicado apenas para o Squid, então o cache pode ter até 1/3 da memória RAM do servidor. Não caia no erro de reservar quase toda a RAM para o cache, pois além do cache, o sistema vai precisar de memória para fazer outras coisas.

O cache no HD pode ser mais generoso, afinal a idéia é que ele guarde todo tipo de arquivos, principalmente os downloads grandes, que demoram para ser baixados. A única limitação neste caso é o espaço livre no HD.

A configuração do cache é feita adicionando mais algumas linhas no arquivo de configuração:

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
```

Abaixo vai mais uma linha, que determina o tamanho máximo dos arquivos que serão guardados no cache feito na memória RAM. O resto vai para o cache feito no HD. O cache na memória é muito mais rápido, mas como a quantidade de RAM é muito limitada, melhor deixa-la disponível para páginas web, figuras e arquivos pequenos em geral. 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 o grosso 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. Sempre é

mais rápido ler a partir do cache do que baixar de novo da web, mesmo que o arquivo seja pequeno.

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
minimum_object_size 0 KB
```

Você pode definir ainda a porcentagem 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 90% cheio:

```
cache_swap_low 90
cache_swap_high 95
```

Depois vem a configuração do tamanho do cache em disco propriamente dita, que 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.

O "2048" indica a quantidade de espaço no HD (em MB) que será usada para o cache. Aumente o valor se você tem muito espaço no HD do servidor e quer que o squid guarde os downloads por muito tempo.

Finalmente, os números 16 256 indicam a quantidade de subpastas que serão criadas dentro do diretório. Por padrão temos 16 pastas com 256 subpastas 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 squid guarda o log de acesso no arquivo `/var/log/squid/access.log`. Este arquivo é usado pelo `sarg` para gerar as páginas com as estatísticas de acesso.

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

Mais uma configuração que você pode querer alterar é o padrão de atualização do cache. Estas três linhas precisam sempre ser usadas em conjunto. Ou seja, você pode alterá-las, mas sempre as três precisam estar presente no arquivo. Eliminando um, o squid ignora as outras duas e usa o default.

Os números indicam o tempo (em minutos) quando o squid irá verificar se um item do cache (uma página por exemplo) foi atualizado, para cada um dos três protocolos.

O primeiro número (o 15) 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 terceiro número (o 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:

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

Depois de adicionar estas configurações todas, o nosso arquivo de configuração já ficará bem maior:

```
http_port 3128
visible_hostname kurumin

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
```

Aqui já temos uma configuração mais completa, incluindo um conjunto de regras de segurança, para que o proxy seja usado apenas a partir da rede local e a configuração do cache, uma configuração adequada para uso numa rede doméstica ou um pequeno escritório por exemplo.

Numa rede maior você provavelmente iria querer adicionar algumas limitações de acesso, limitando o acesso a algumas páginas, criando um sistema de autenticação ou limitando o uso com base no horário por exemplo.

Adicionando restrições de acesso

Num ambiente de trabalho, a idéia é que os funcionários usem a internet para comunicação, pesquisa e outras funções relacionadas ao que estão fazendo. Algumas empresas permitem que acessem os e-mails pessoais e coisas assim, mas sempre até um certo limite. Seu chefe não vai gostar se começarem a passar a maior parte do tempo no Orkut por exemplo.

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.

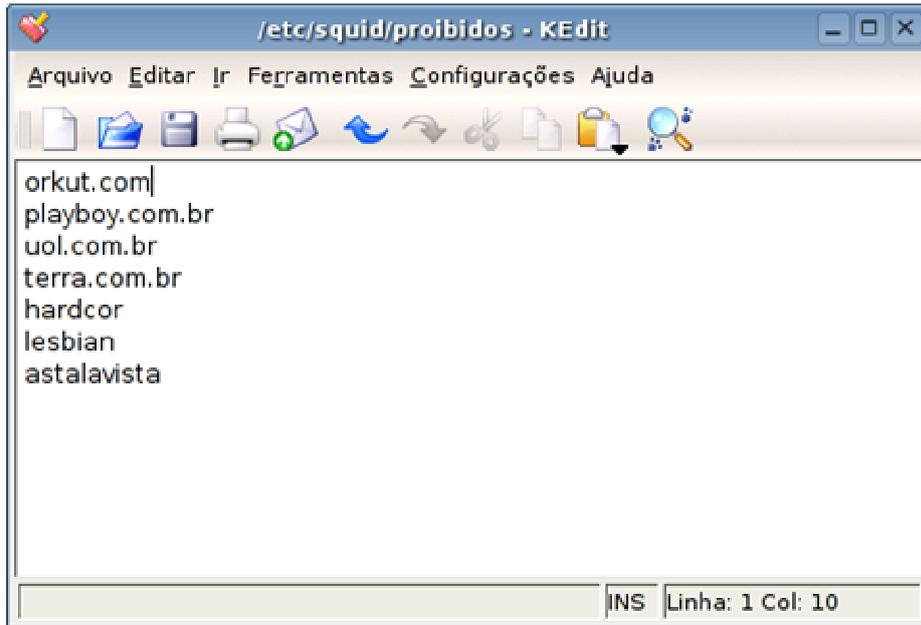
Se você bloquear a palavra "sexo" por exemplo, qualquer site ou artigo que mencione a palavra será bloqueado. Ao bloquear por palavras você deve tentar ser específico, bloqueando apenas jargões e expressões que são encontradas apenas nos sites que você pretende bloquear.

Para adicionar o filtro de palavras, adicione as linhas:

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

Aqui 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 deve conter as palavras e domínios bloqueados, um por linha:



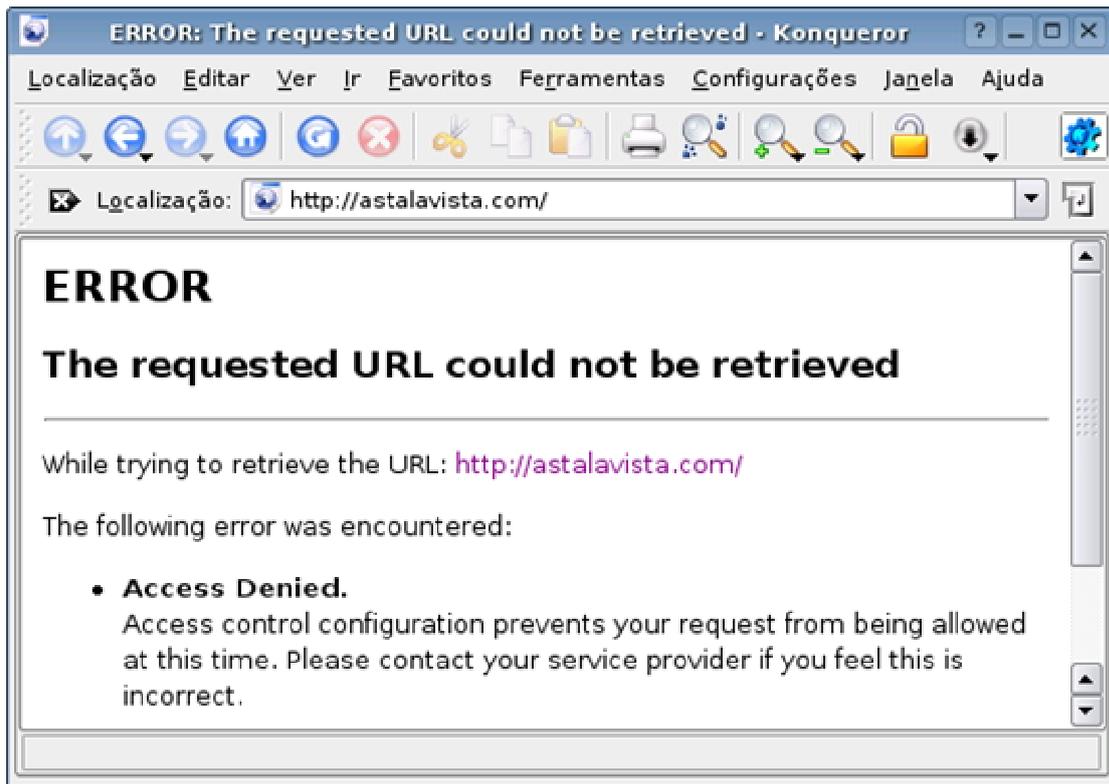
Depois criar o arquivo e adicionar as duas linhas no arquivo de configuração do squid, reinicie o serviço com um:

```
# service squid restart
```

ou

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

A partir daí o bloqueio de palavras entra em ação e os clientes passam a ver uma mensagem de erro ao tentar acessar a página:



Este filtro de palavras pode levar a alguns erros inesperados. Se você o usa para bloquear o site "orkut.com" por exemplo, os usuários não conseguirão ler por exemplo um artigo da Superinteressante que cita o endereço do site.

Existe uma opção mais adequada para o bloqueio de domínios, que é criar uma acl usando o parâmetro "dstdomain". Veja um exemplo:

```
acl bloqueados dstdomain orkut.com playboy.abril.com.br astalavista.box.sk
http_access deny bloqueados
```

Aqui eu criei uma acl chamada "**bloqueados**" que contém os endereços "orkut.com" "playboy.abril.com.br" e "astalavista.box.sk" e me seguida incluí a regra "http_access deny bloqueados" que bloqueia o acesso a eles.

Aqui estamos sendo mais específicos. Ao invés de bloquear o termo "orkut.com" o squid vai bloquear apenas o acesso ao domínio em questão.

Existe uma última ressalva: alguns sites, como o orkut.com podem ser acessados tanto com o www quanto sem. Para o squid, "www.orkut.com" e "orkut.com" são duas coisas diferentes. Bloqueando o "orkut.com" os usuários ainda conseguirão acessar o site através do "www.orkut.com". Para bloquear ambos, é preciso incluir as duas possibilidades dentro da regra, como em:

```
acl bloqueados dstdomain orkut.com www.orkut.com playboy.abril.com.br
http_access deny bloqueados
```

Você pode incluir quantos domínios quiser dentro da regra, basta separá-los por espaço e deixar tudo na mesma linha.

Não existe problema em combinar a regra que cria o filtro de palavras com esta que filtra baseado no domínio, você pode apelar para uma ou outra de acordo com a situação.

Depois de adicionar as novas regras, nosso arquivo de configuração ficaria assim:

```
http_port 3128
visible_hostname kurumin

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 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
```

Veja que coloquei as duas regras antes do "http_access allow redelocal", que abre tudo para a rede local. Como o squid processa as regras seqüencialmente, as páginas que forem bloqueadas pelas duas regras não chegarão a passar pela seguinte.

Bloqueando por horário

Estas regras fazem com que o proxy recuse conexões feitas dentro de determinados horários. Você pode definir regras períodos específicos e combiná-las para bloquear todos os horários em que você não quer que o proxy seja usado.

Para que o proxy bloqueie acessos feitos entre meia-noite e 6:00 da manhã e no horário de almoço por exemplo, você usaria as regras:

```
acl madrugada time 00:00-06:00
http_access deny madrugada
```

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

Estas regras iriam novamente antes da regra "http_access allow redelocal" no arquivo de configuração.

Agora imagine que você quer fazer diferente. Ao invés de bloquear o acesso na hora de almoço, você quer deixar o proxy aberto, para quem quiser ir no orkut ou acessar os e-mails poder fazer isso fora do horário de trabalho. Neste caso você usaria uma regra como:

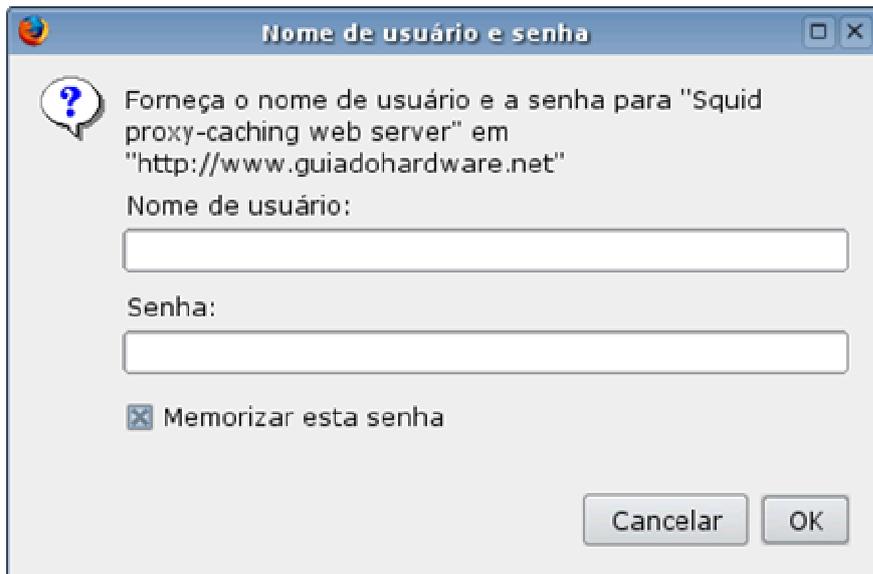
```
acl almoco time 12:00-14:00
http_access allow almoco
```

Esta regra entraria no arquivo de configuração antes das regras "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.

Proxy com 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 e auditar os acessos em caso de necessidade.

Quase todos os navegadores oferecem a opção de salvar a senha. Não seria muito legal se o usuário tivesse que ficar digitando toda hora... ;-)



Para ativar a autenticação você vai precisar de u 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:

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

Cadastre os logins usando o comando:

```
# htpasswd /etc/squid/squid_passwd kurumin  
(onde o "kurumin" é o usuário que está sendo adicionado)
```

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
```

O `"/usr/lib/squid/ncsa_auth"` é a localização da biblioteca responsável pela autenticação. Eventualmente, ela pode estar numa pasta diferente dentro da distribuição que estiver usando. Neste caso use o "locate" ou a busca do KDE para encontrar o arquivo e altere a linha indicando a localização correta.

Estas três linhas criam uma acl chamada "autenticados" (poderia ser outro nome), que contém os usuários que se autenticarem usando um login válido.

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 e o firewall 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.

Outra vantagem é que este recurso permite usar o proxy sem precisar configurar manualmente o endereço em cada estação. Basta usar o endereço IP do servidor rodando o proxy como gateway da rede.

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
```

O "eth1" na segunda linha, indica a placa que está conectada na Internet e será compartilhada. Você pode checar a configuração da rede usando o comando "ifconfig" (como root).

Nem todas as distribuições instalam o iptables por padrão. No Mandrake por exemplo, pode ser necessário rodar primeiro um "urpmi iptables".

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
```

O "eth0" neste quarto comando indica a placa da rede local, onde o proxy recebe as requisições dos outros micros da rede e o "3128" indica a porta usada pelo squid.

Adicione os quatro comandos no final do arquivo **/etc/rc.d/rc.local** ou **/etc/init.d/bootmisc.sh** (no Debian) para que eles sejam executados durante o boot.

Finalmente, você precisa adicionar as seguintes linhas no final do arquivo 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
```

Em resumo, você vai ter a conexão compartilhada via Nat no servidor e configurará os clientes para acessar através dela, colocando o servidor como gateway da rede. Ao ativar o proxy transparente, a configuração dos clientes continua igual, a única diferença é que agora todo o tráfego da porta 80 passará obrigatoriamente pelo servidor Squid.

Isso permite que você se beneficie do log dos acessos e do cache feito pelo proxy, sem ter que se sujeitar às desvantagens de usar um proxy, como ter que configurar manualmente cada estação.

Depois de adicionar a regra que libera o acesso na hora do almoço, ativar a autenticação e o proxy transparente, nosso arquivo vai ficar assim:

```
http_port 3128
visible_hostname kurumin

# Configuração do cache
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

# Localização do log de acessos do Squid
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

# Libera acessos na hora do almoço
acl almoco time 12:00-14:00
http_access allow almoco

# Filtros por palavras e por dominios
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

# Autenticação dos usuários
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/squid_passwd
acl autenticados proxy_auth REQUIRED
http_access allow autenticados

# Libera para a rede local
acl redelocal src 192.168.1.0/24
http_access allow localhost
http_access allow redelocal

# Bloqueia acessos externos
http_access deny all

# Proxy transparente
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Usando o Sarg para monitorar o acesso

O Sarg é um interpretador de logs para o Squid, assim como o Webalizer e 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.

A partir daí você pode acompanhar as páginas que estão sendo acessadas, mesmo que não exista nenhum filtro de conteúdo e tomar as medidas cabíveis em casos de abuso. Todos sabemos que os filtros de conteúdo nunca são completamente eficazes, eles sempre bloqueiam algumas páginas úteis e deixam passar muitas páginas impróprias. Se você tiver algum tempo para ir acompanhando os logs, a inspeção manual é sempre o método mais eficiente.

A partir daí você pode ir fazendo um trabalho incremental, de ir bloqueando uma a uma páginas onde os usuários perdem muito tempo, ou fazer algum trabalho educativo, explicando que os acessos estão sendo monitorados e estabelecendo algum tipo de punição para quem abusar.

Aqui está um exemplo do relatório gerado pelo Sarg. Por padrão ele gera um conjunto de páginas html dentro da pasta `/var/www/squid-reports/` (ou `/var/www/html/squid/`, no Mandrake) 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.

Kurumin, log de acessos através do proxy
 Período: 2004Oct02-2004Oct02
 Ordem: BYTES, reverse
 Topuser Relatório

[REGRA](#) Relatório
[Sites & Users](#) Relatório
[SQUIDGUARD](#) Relatório
[Falha de autenticação](#) Relatório

NUM	USUÁRIO	CONEXÃO	BYTES	%BYTES	IN-CACHE	OUT	TEMPO GASTO	MILISEG	%TEMPO
1	data/hora192.168.0.2	266	1,314,660	81.81%	1.18%	98.82%	00:09:19	559,519	76.63%
2	data/hora kurumin	46	287,118	17.87%	14.85%	85.15%	00:02:39	159,650	21.87%
3	data/hora127.0.0.1	11	5,199	0.32%	0.00%	100.00%	00:00:10	10,985	1.50%
TOTAL			3231,606,977	3.62%	96.39%		00:12:10	730,154	
MÉDIA			107				00:04:03	243,384	

Gerado por [sarg-1.4.1](#) 25Apr2003 em Oct/02/2004 09:04

Página carregada.

O Sarg é incluído na maioria das distribuições atuais, em alguns casos instalado por padrão junto com o Squid.

No Debian e derivados ele pode ser instalado com um:

```
# apt-get install sarg
```

No Mandrake um "**urpmi sarg**" já resolve.

Depois de instalado, basta chamar o comando "**sarg**" (como root) para que os relatórios sejam geradas automaticamente a partir do log do squid.

O Sarg não é um daemon que fica residente, 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 cosméticas no arquivo de configuração do Sarg, que é o `/etc/sarg/sarg.conf` (no Mandrake) ou `/etc/squid/sarg.conf` (no Debian).

O arquivo é auto explicativo, nele você pode alterar os diretórios padrão, alterar o layout da Outro recurso interessante é o envio de uma cópia do relatório por e-mail sempre que o sarg for executado.

Samba, parte 1: Instalação e configuração usando o swat

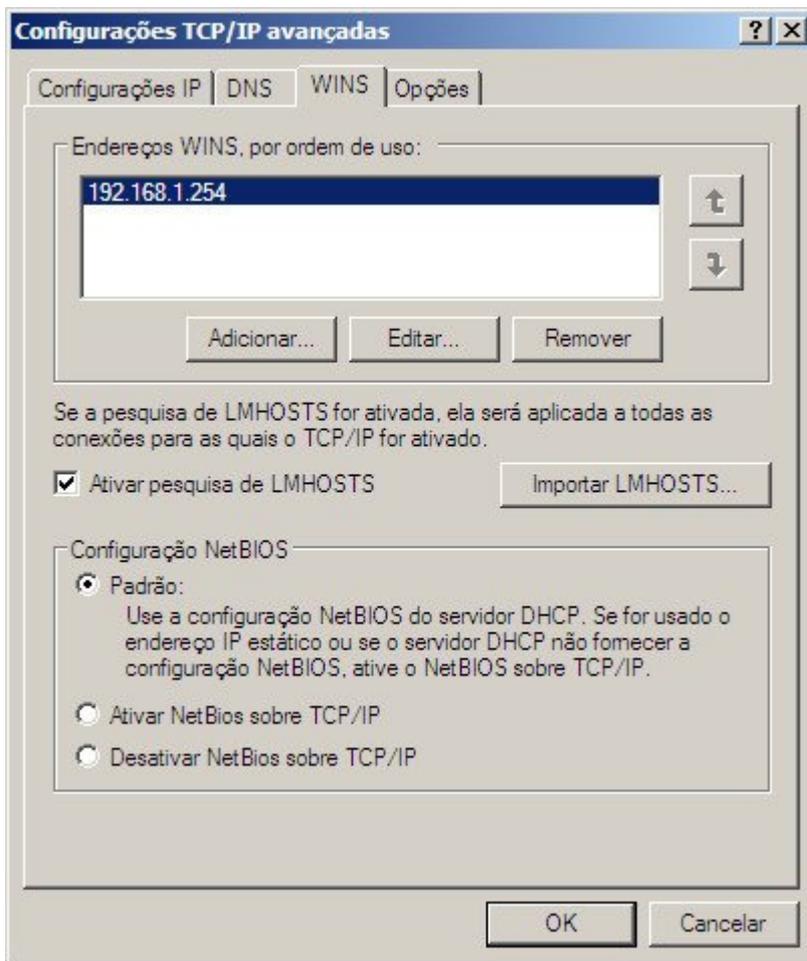
A necessidade de compartilhar arquivos e impressoras motivou o aparecimento das primeiras redes (ainda na década de 70) e continua sendo uma necessidade comum. Mesmo para fazer um simples backup armazenado remotamente, é necessário configurar algum tipo de compartilhamento de arquivos.

Existem diversas formas de disponibilizar arquivos, incluindo o NFS, o FTP, o SFTP e até mesmo um servidor web, que pode ser usado para compartilhar a pasta contendo os arquivos e aceitar uploads através de um script em PHP. Entretanto, quando falamos em redes locais, o protocolo mais usado é o CIFS (Common Internet File System), que é o protocolo usado para compartilhar arquivos e impressoras em redes Microsoft.

O nome "CIFS" pode soar estranho à primeira vista, mas ele nada mais é do que a mais nova versão do protocolo SMB, usada a partir do Windows 2000. A história do SMB e do CIFS começa em 1984, quando a IBM criou o protocolo NetBIOS (Network Basic Input Output), um protocolo para troca de mensagens entre máquinas da rede, originalmente desenvolvido para servir como uma extensão do BIOS da placa-mãe, oferecendo recursos de rede. Em 1985 o protocolo foi expandido, dando origem ao protocolo NetBEUI, que foi durante muito tempo o principal protocolo usado em redes locais, antes da popularização do TCP/IP.

O SMB (Server Message Block) veio mais tarde, junto com o Windows 3.11. O protocolo SMB governa o compartilhamento de arquivos e impressoras em redes Microsoft, incluindo a navegação na rede, o estabelecimento de conexões e a transferência de dados. O SMB utiliza o NetBIOS para a troca de mensagens entre os hosts e inclui uma versão atualizada do NetBIOS, que roda sobre o TCP/IP. Acessando as

propriedades do protocolo TCP/IP dentro das configurações de rede de uma máquina com o Windows XP, você pode ver que ele continua presente, com o objetivo de manter compatibilidade com as versões anteriores do Windows:



O problema com o NetBIOS é que ele depende do uso intensivo de pacotes de broadcast e de pacotes UDP. O CIFS é a evolução natural do SMB, que inclui diversos novos recursos, abandona o uso do NetBIOS e passa a utilizar uma única porta TCP (445) no lugar das três portas (137 UDP, 138 UDP e 139 TCP) utilizadas pelo SMB.

O Samba é justamente uma implementação das mesmas funções para sistemas Unix, incluindo não apenas o Linux, mas também o BSD, Solaris, OS X e outros primos. Ele começou como uma implementação do protocolo SMB e depois foi sucessivamente expandido e atualizado, de forma a incorporar suporte ao CIFS.

O Samba começou no final de 1991, de forma acidental. Andrew Tridgell, um Australiano que na época era estudante do curso de PhD em Ciências da Computação da Universidade Nacional da Austrália. Ele precisava rodar um software da DEC chamado "eXcursion", que trabalhava em conjunto com o Patchworks, um software de compartilhamento de arquivos que utilizava um protocolo obscuro, que mais tarde se revelou uma implementação do protocolo SMB desenvolvida pela DEC.

Como todo bom hacker, ele decidiu estudar o protocolo e assim desenvolver um servidor que pudesse rodar em seu PC. Ele desenvolveu então um pequeno programa, chamado clockspy, que era capaz de examinar o tráfego da rede, capturando as mensagens

enviadas pelo cliente e as respostas do servidor. Com isso, ele foi rapidamente capaz de implementar o suporte às principais chamadas e a desenvolver um programa servidor, que era capaz de conversar com os clientes rodando o Patchworks.

Pouco depois, em janeiro de 1992 ele disponibilizou o "Server 0.1" no servidor da Universidade, que foi rapidamente seguido por uma versão aprimorada, o "Server 0.5". Este arquivo ainda pode ser encontrado em alguns dos FTPs do <http://samba.org>, com o nome "server-0.5".

Esta versão inicial rodava sobre o MS-DOS. Depois de um longo período de hibernação, o software foi portado para o Linux, dando versão à versão seguinte (1.5), que foi lançada apenas em dezembro de 1993 e passou a se chamar "smbserver". O nome continuou sendo usado até abril de 2004, quando foi finalmente adotado o nome definitivo.

O nome "Samba" surgiu a partir de uma simples busca dentro do dicionário Ispell por palavras que possuíssem as letras S, M e B, de "Server Message Blocks", posicionadas nessa ordem. A busca retornou apenas as palavras "salmonberry", "samba", "sawtimber" e "scramble", de forma que a escolha do nome acabou sendo óbvia.

Uma curiosidade é que não existiu um "Samba 1.0", pois a primeira versão a utilizar o nome "Samba" foi a 1.6.05, que foi a sucessora imediata do "smbserver 1.6.4".

O projeto começou a se tornar popular a partir da versão 1.6.09 (lançada pouco depois), que foi a primeira a trazer suporte ao controle de acesso com base nos logins de usuário (assim como o Windows NT), enquanto as versões anteriores suportavam apenas o controle de acesso com base no compartilhamento (assim como no Windows 3.11 e 95), onde a única opção de segurança era usar uma senha de acesso para os compartilhamentos. A partir daí o projeto não parou de crescer, atraindo um número crescente de usuários e desenvolvedores, até se transformar no monstro sagrado que é hoje.

Estes dois links contam um pouco mais sobre a história do Samba, desde as primeiras versões:

<http://www.samba.org/samba/docs/10years.html>

<http://www.rxn.com/services/faq/smb/samba.history.txt>

Em 94 a Microsoft liberou as especificações do SMB e do NetBios, o que permitiu que o desenvolvimento do Samba desse um grande salto, tanto em recursos quanto em compatibilidade, passando a acompanhar os novos recursos adicionados ao protocolo da Microsoft, que mais tarde novamente deixou de ser aberto.

Hoje, além de ser quase 100% compatível com os recursos de rede do Windows 98, NT, 2000 e XP, o Samba é reconhecido por ser mais rápido que o próprio Windows na tarefa de servidor de arquivos.

Um dos pontos fortes do Samba é que o projeto foi todo desenvolvido sem precisar apelar para qualquer violação de patentes. Todas as chamadas (com exceção das que a Microsoft tornou públicas em 94) foram implementadas monitorando as transmissões de dados através da rede, que os desenvolvedores costumam chamar de "French Café technique". Dentro do exemplo, seria como aprender francês sentando-se em um café e passar a prestar atenção nas conversas e a partir daí ir aprendendo novas palavras e expressões e

situações onde elas podem ou não ser usadas. É um trabalho bastante detalhista e tedioso, que demanda um grande esforço e resulta em avanços graduais, mas se executado por anos a fio, como no caso do Samba, que começou a ser desenvolvido em 1991 (já que a primeira versão pública foi disponibilizada em janeiro de 1992) resulta em conquistas surpreendentes.

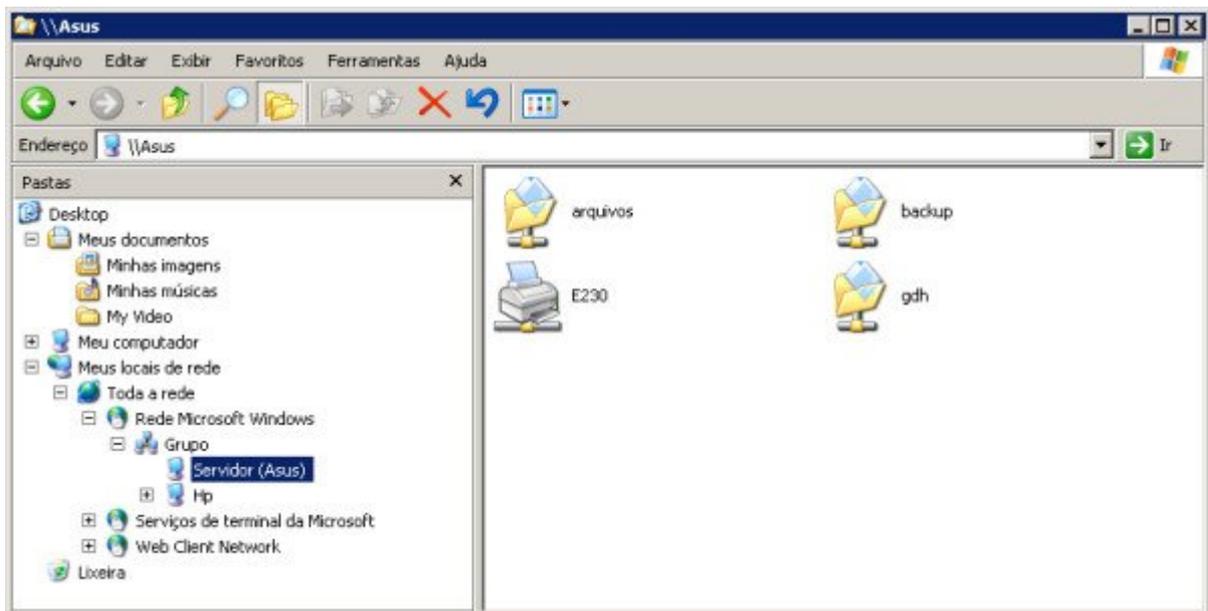
Isso torna o Samba virtualmente a ações legais relacionadas a quebras de patentes ou problemas similares, já que o software é inteiramente baseado em observação e no uso de especificações públicas.

Uma curiosidade é que a existência do Samba permitiu que a Microsoft conseguisse colocar PCs rodando o Windows em muitos nichos onde só entravam Workstations Unix, já que com o Samba os servidores Unix existentes passaram a ser compatíveis com as máquinas Windows. Ou seja: até certo ponto o desenvolvimento Samba foi vantajoso até mesmo para a Microsoft.

Quase tudo que você pode fazer usando um servidor Windows, pode ser feito também através do Samba, com uma excelente segurança e confiabilidade e com um desempenho em muitas situações bastante superior ao de um servidor Windows com a mesma configuração.

O Samba é uma solução bastante completa e flexível para uso em redes locais, pois inclui várias opções de segurança e, além de compartilhar arquivos, permite também compartilhar impressoras e centralizar a autenticação dos usuários, atendendo tanto a clientes Windows, quanto clientes Linux.

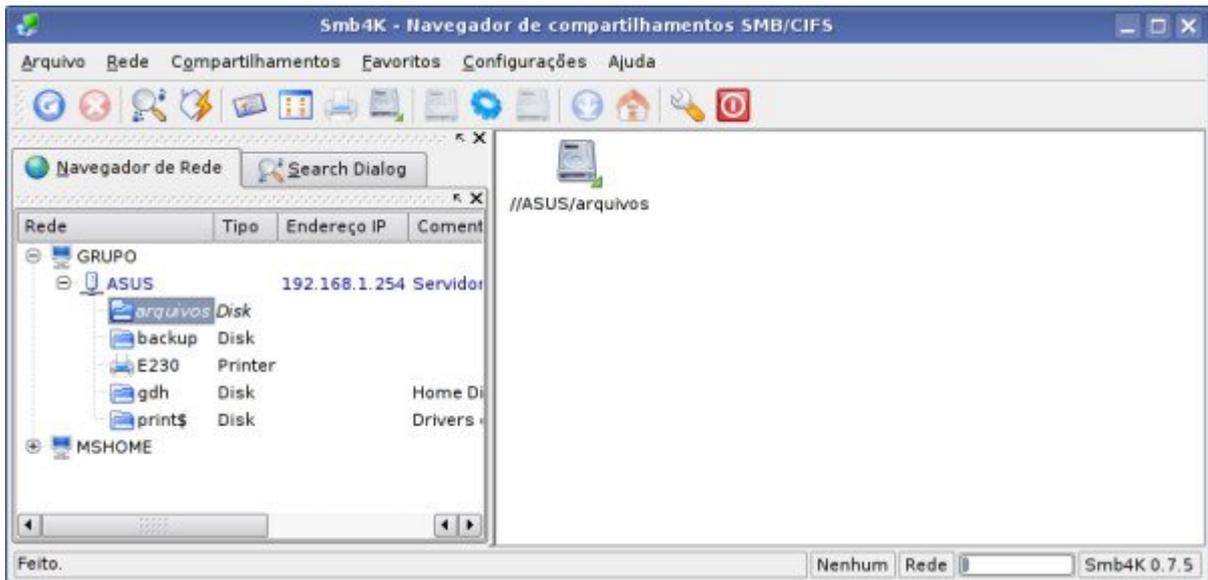
Para os clientes Windows, o servidor Samba aparece no ambiente de redes exibindo os compartilhamentos disponíveis, exatamente da mesma forma que um servidor Windows:



Os compartilhamentos podem ser acessados da forma tradicional e inclusive mapeados. No caso dos compartilhamentos de impressoras, é possível inclusive utilizar o Point-and-Print, onde os clientes obtêm os drivers de impressão diretamente a partir do servidor e a impressora fica disponível com apenas dois cliques.

Além de compartilhar arquivos e impressoras, o servidor Samba pode atuar como um PDC, autenticando os usuários da rede. Existem também diversas opções de segurança, que permitem restringir o acesso aos compartilhamentos.

Naturalmente, o servidor pode ser acessado de forma simples também nos clientes Linux da rede. As versões recentes do Konqueror e do Nautilus incorporam plugins que permitem acessar os compartilhamentos de forma bastante prática (experimente digitar `smb://endereço_do_servidor` na barra de endereços do Konqueror) e existem também clientes especializados, como o Smb4k:



O Samba é dividido em dois módulos, o servidor Samba propriamente dito e o "smbclient", o cliente que permite acessar compartilhamentos em outras máquinas. Usando o Samba, o servidor Linux se comporta exatamente da mesma forma que uma máquina Windows, compartilhando arquivos e impressoras e executando outras funções, como autenticação de usuários. Você pode configurar o Samba até mesmo para tornar-se um controlador de domínio.

Instalando

Como comentei a pouco, o Samba é dividido em dois módulos. O servidor propriamente dito e o cliente, que permite acessar compartilhamentos em outras máquinas (tanto Linux quanto Windows). Os dois são independentes, permitindo que você mantenha apenas o cliente instalado num desktop e instale o servidor apenas nas máquinas que realmente forem compartilhar arquivos. Isso permite melhorar a segurança da rede de uma forma geral.

Os pacotes do Samba recebem nomes um pouco diferentes nas distribuições derivadas do Debian e no Fedora e outras distribuições derivadas do Red Hat. Veja:

Pacote Debian Fedora

Servidor: samba samba

Cliente: smbclient samba-client

Documentação samba-doc samba-doc

Swat: swat samba-swat

Para instala-lo no Debian ou Ubuntu, por exemplo, você usaria:

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

O script de instalação faz duas perguntas. A primeira é se o servidor deve rodar em modo daemon ou sob o inetd. Responda "**daemons**" para quer o servidor rode diretamente. Isso garante um melhor desempenho, melhor segurança e evita problemas diversos de configuração relacionados ao uso do inetd, serviço que está entrando em desuso.

Em seguida ele pergunta: "Gerar a base de dados para senhas /var/lib/samba/passdb.tdb?". É importante responder que "**Sim**", para que ele crie o arquivo onde serão armazenadas as senhas de acesso. Como explica o script, "*Caso você não o crie, você terá que reconfigurar o samba (e provavelmente suas máquinas clientes) para utilização de senhas em texto puro*", o que é um procedimento trabalhoso, que consiste em modificar chaves de registro em todas as máquinas Windows da rede e modificar a configuração de outros servidores Linux. Muito mais fácil responder "Sim" e deixar que ele utilize senhas encriptadas, que é o padrão. :)

Lembre-se de que você deve instalar todos os pacotes apenas no servidor e em outras máquinas que forem compartilhar arquivos. O Swat ajuda bastante na etapa de configuração, mas ele é opcional, pois você pode tanto editar manualmente o arquivo smb.conf, quanto usar um arquivo pronto, gerado em outra instalação. Nos clientes que forem apenas acessar compartilhamentos de outras máquinas, instale apenas o cliente.

O Fedora inclui mais um pacote, o "system-config-samba", um utilitário de configuração rápida, que permite criar e desativar compartilhamentos de forma bem prática. Outro configurador rápido é o módulo "Internet & Rede > Samba", disponível no Painel de Controle do KDE. Neste tutorial abordo apenas o swat, que é o configurador mais completo, mas você pode lançar mão destes dois utilitários para realizar configurações rápidas.

Com os pacotes instalados, use os comandos:

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

... para iniciar e parar o serviço. 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
```

Para reativá-lo mais tarde, use:

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

No **Fedora** e **Mandriva**, os comandos para iniciar e parar o serviço são:

```
# service smb start
# service smb stop
```

Para desabilitar o carregamento durante o boot, use o "**chkconfig smb off**" e, para reativar, use o "**chkconfig smb on**". Note que, em ambos, o pacote de instalação se chama "samba", mas o serviço de sistema chama-se apenas "smb".

É sempre recomendável utilizar os pacotes que fazem parte da distribuição, que são compilados e otimizados para o sistema e recebem atualizações de segurança regularmente. De qualquer forma, você pode encontrar também alguns pacotes compilados por colaboradores no http://samba.org/samba/ftp/Binary_Packages/, além do código fonte, disponível no <http://samba.org/samba/ftp/stable/>. Ao instalar a partir do fonte o Samba é instalado por default na pasta "usr/local/samba", com os arquivos de configuração na pasta "/usr/local/samba/lib".

Este texto é baseado no Samba 3 que, enquanto escrevo, é a versão estável, recomendada para ambientes de produção. O Samba 3 trouxe suporte ao Active Directory, passou a ser capaz de atuar como PDC, trouxe muitas melhorias no suporte a impressão e inúmeras outras melhorias em relação à série 2.x.

O Samba 3.0.0 foi lançado em setembro de 2003, ou seja, a mais de 4 anos. Comparado com os ciclos de desenvolvimento das distribuições Linux, que são em sua maioria atualizadas a cada 6 ou 12 meses, 4 anos podem parecer muita coisa, mas se compararmos com os ciclos de desenvolvimento de novas versões do Windows, por exemplo, os ciclos parecem até curtos :). Para efeito de comparação, o Samba 2, o major release anterior foi lançado em 1999 e o Samba 4 está em estágio de desenvolvimento, ainda sem previsão de conclusão.

Por ser um software utilizado em ambientes de produção, novas versões do Samba são exaustivamente testadas antes de serem consideradas estáveis e serem oficialmente lançadas. Graças a isso, é muito raro o aparecimento de bugs graves e, quando acontecem, eles costumam ser corrigidos muito rapidamente.

Naturalmente, as versões de produção continuam sendo atualizadas e recebendo novos recursos. Entre o Samba 3.0.0 lançado em 2003 e o Samba 3.0.24 incluído no Debian Etch, por exemplo, foram lançadas nada menos do que 28 minor releases intermediários. Se tiver curiosidade em ler sobre as alterações em cada versão, pode ler o change-log de cada versão no: <http://samba.org/samba/history/>.

Você pode verificar qual é a versão do Samba instalada usando o comando "smbd -V", como em:

```
# smbd -V
Version 3.0.24
```

Ao usar qualquer distribuição atual, muito provavelmente Você encontrará o Samba 3.0.23 ou superior. Se por acaso você estiver usando alguma distribuição muito antiga, que ainda utilize uma versão do Samba anterior à 3.0.0, recomendo que atualize o sistema, já que muitos dos recursos que cito ao longo do texto, sobretudo o uso do Samba como PDC não funcionam nas versões da série 2.x.

Cadastrando os usuários

Depois de instalado, o próximo passo é cadastrar os logins e senhas dos usuários que terão acesso ao servidor. Esta é uma peculiaridade do Samba: ele roda como um programa sobre o sistema e está subordinado às permissões de acesso deste. Por isso, ele só pode dar acesso para usuários que, além de estarem cadastrados no Samba, também estão cadastrados no sistema.

Existem duas abordagens possíveis. Você pode criar usuários "reais", usando o comando **adduser** ou um utilitário como o "**user-admin**" (disponível no Fedora e no Debian, através do pacote `gnome-system-tools`). Ao usar o `adduser`, o comando fica:

```
# adduser maria
```

Uma segunda opção é criar usuários "castrados", 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 a manter a conta desativada até segunda ordem:

```
# adduser --disabled-login --no-create-home maria
```

Isso cria uma espécie de usuário fantasma que, para todos os fins, existe e pode acessar arquivos do sistema (de acordo com as permissões de acesso), mas que, por outro lado, não pode fazer login (nem localmente, nem remotamente via SSH), nem possui diretório `home`.

Uma dica é que no **Fedora** (e outras distribuições derivadas do Red Hat), você só consegue usar o comando `case` logue-se como `root` usando o comando "**su -**" ao invés de simplesmente "`su`". A diferença entre os dois é que o "`su -`" ajusta as variáveis de ambiente, incluindo o `PATH`, ou seja, as pastas onde o sistema procura pelos executáveis usados nos comandos. Sem isso, o Fedora não encontra o executável do `adduser`, que vai na pasta `"/usr/sbin"`.

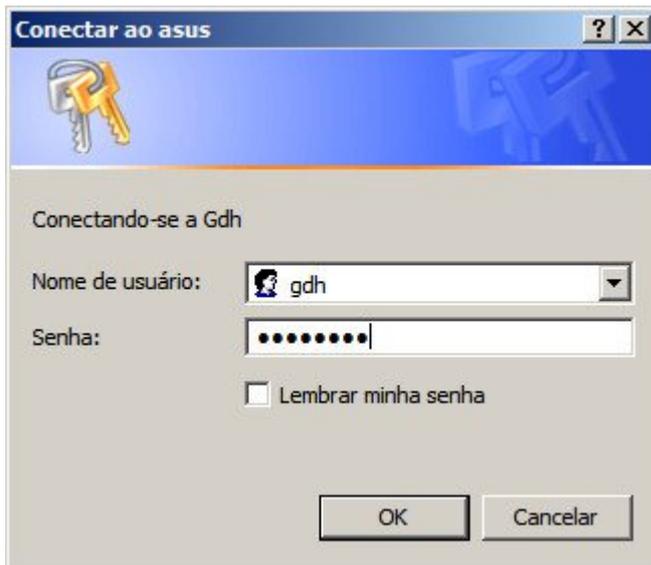
Os parâmetros suportados pelo `adduser` também são um pouco diferentes. O padrão já é criar um login desabilitado (você usa o comando "`passwd usuário`" para ativar) e, ao invés do "`--no-create-home`", usa a opção "`-M`". O comando (no Fedora) fica, então:

```
# adduser -M maria
```

De qualquer uma das duas formas, depois de criar os usuários no sistema você deve cadastrá-los no Samba, usando o comando "**smbpasswd -a**", como em:

```
# smbpasswd -a maria
```

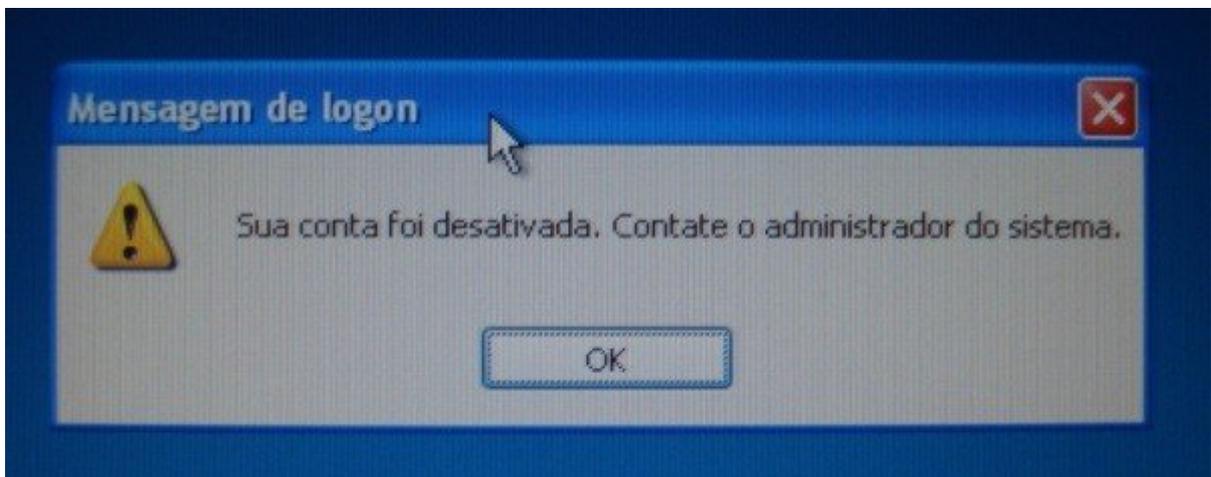
Se você mantiver os logins e senhas sincronizados com os usados pelos usuários nos clientes Windows, o acesso aos compartilhamentos é automático. Caso os logins ou senhas no servidor sejam diferentes, o usuário precisará fazer login ao acessar:



Um detalhe importante é que, ao usar clientes Windows 95/98/ME, 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 (Painel de controle > Redes).

Para desativar temporariamente um usuário, sem removê-lo do sistema (como em situações onde um funcionário sai de férias, ou um aluno é suspenso), você pode usar o parâmetro "-d" (disable) do smbpasswd, como em:

```
# smbpasswd -d maria
```



Para reativar a conta posteriormente, use o parâmetro "-e" (enable), como em:

```
# smbpasswd -e maria
```

Se, por outro lado, você precisar remover o usuário definitivamente, use o parâmetro "-x" (exclude), seguido pelo comando "deluser", que remove o usuário do sistema, como em:

```
# smbpasswd -x maria  
# deluser maria
```

Depois de criados os logins de acesso, falta agora apenas configurar o Samba para se integrar à rede e compartilhar as pastas desejadas, trabalho facilitado pelo **Swat**. A segunda opção é editar manualmente o arquivo de configuração do Samba, o `"/etc/samba/smb.conf"`, como veremos mais adiante. As opções que podem ser usadas no arquivo são as mesmas que aparecem nas páginas do Swat, de forma que você pode até mesmo combinar as duas coisas, configurando através do Swat e fazendo pequenos ajustes manualmente, ou vice-versa.

Usando o Swat

O Swat é um utilitário de configuração via web, similar ao encontrado nos modems ADSL. Isso permite que ele seja acessado remotamente e facilita a instalação em servidores sem o ambiente gráfico instalado. Esta mesma abordagem é utilizada por muitos outros utilitários, como o Webmin.

Manter o ambiente gráfico instalado e ativo em um servidor dedicado é considerado um desperdício de recursos, por isso os desenvolvedores de utilitários de configuração evitam depender de bibliotecas gráficas. Desse modo, mesmo distribuições minimalistas podem incluí-los. No caso de redes de pequeno ou médio porte, você pode até mesmo usar uma máquina antiga como servidor de arquivos, fazendo uma instalação minimalista do Debian, Ubuntu ou outra distribuição e instalando o Samba e o Swat em modo texto.

Como facilitador, o Swat acaba sendo uma faca de dois gumes, pois ao mesmo tempo em que facilita a configuração, por ser uma ferramenta visual e dispensar a edição manual do arquivo, ele complica, por oferecer um grande número de opções específicas ou obsoletas. Vamos então aprender como fazer uma configuração básica usando o Swat e depois nos aprofundar na configuração do Samba editando o `smb.conf` manualmente. Se preferir, você pode ir diretamente para o tópico seguinte.

No Debian, Slackware e também no Gentoo, o Swat é inicializado através do **inetd**. O `inetd` tem a função de monitorar 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 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
```

No caso do Ubuntu, o `inetd` não vem instalado por padrão. A documentação recomenda usar o **xinetd** no lugar dele, o que é uma boa deixa para falar um pouco sobre as diferenças de configuração entre os dois serviços. O `xinetd` tem a mesma função do `inetd` ou seja, carregar serviços sob demanda, mas é mais recente e um pouco mais seguro, de forma que tem sido mais usado.

O primeiro passo para instalar o Swat no Ubuntu seria instalar o swat e o xinetd usando o apt-get:

```
# apt-get install swat xinetd
```

Ao invés de usar um único arquivo central, no xinetd é utilizada uma pasta com arquivos de configuração separados para cada serviço em vez de um arquivo único como no caso do inetd. Para ativar o swat, é necessário criar o arquivo `"/etc/xinetd.d/swat"`, com o seguinte conteúdo:

```
service swat
{
port = 901
socket_type = stream
wait = no
user = root
server = /usr/sbin/swat
log_on_failure += USERID
disable = no
}
```

Depois de criado o arquivo, reinicie o serviço e o Swat ficará disponível.

```
# /etc/init.d/xinetd restart
```

Nas distribuições derivadas do Red Hat, o Swat também é inicializado através do xinetd. Para ativá-lo depois da instalação, use os comandos:

```
# chkconfig swat on
# service xinetd restart
```

Em caso de problemas, abra o arquivo `"/etc/xinetd.d/swat"` e substitua a linha `"disable = yes"` (caso presente) por `"disable = no"` e reinicie novamente o serviço xinetd. No Fedora, você pode também reiniciar os serviços usando o utilitário `"systemconfig-services"`, que funciona como uma interface gráfica para o comando `"service"`.

Como pode ver, devido às diferenças de configuração entre as distribuições e o uso do xinetd/inetd, ativar o swat pode ser um pouco mais complicado do que ativar outros serviços, embora o Samba propriamente dito não dependa dele para fazer seu trabalho.

Para acessar o Swat localmente, basta abrir o Firefox ou outro Browser disponível e acessar o endereço **`http://localhost:901`**. No prompt de login, forneça a senha de root (do sistema) para acessar. As credenciais do root são necessárias para que o Swat possa alterar os arquivos de configuração, reiniciar os serviços e outras operações que ficam disponíveis apenas para o root. No caso do Ubuntu, você pode definir a senha de root usando o comando `"sudo passwd"`.

Ao configurar um servidor remotamente, ou ao instalar o Samba/Swat em um servidor sem o ambiente gráfico instalado, você pode acessar o swat remotamente, a partir de qualquer máquina rede. Abra o navegador e acesse o endereço `"http://ip-do-servidor:901"`, como em: `http://192.168.1.1:901`.

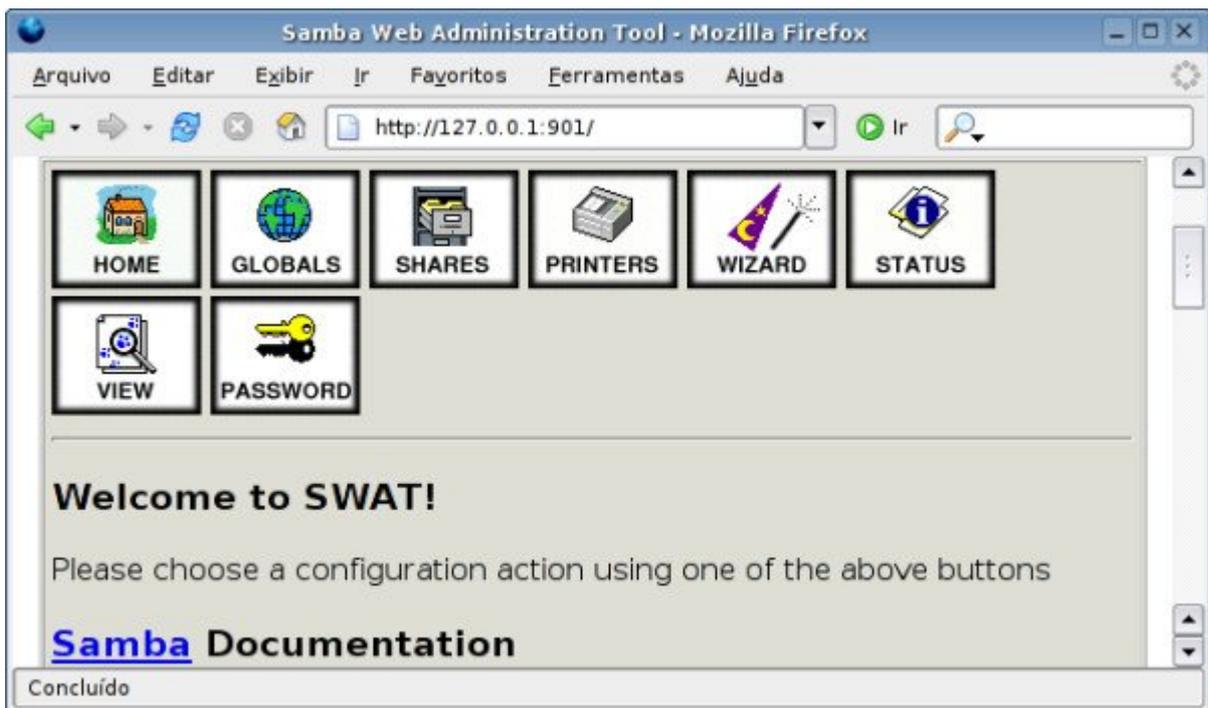
Uma observação é que o Swat não utiliza encriptação, o que é uma temeridade do ponto de vista da segurança, já que alguém poderia capturar a senha sniffando a rede. Você pode evitar isso criando um túnel seguro usando o SSH e acessando o Swat através dele. Para isso, é preciso apenas que o SSH esteja ativado no servidor. Para criar o túnel, use o comando:

```
# ssh -f -N -L901:192.168.1.1:901 -l login 192.168.1.1
```

... onde o "192.168.1.1" é o endereço IP do servidor, o "901" é a porta do Swat e o "login" é a sua conta no servidor. Este comando cria um túnel encriptado entre a porta 901 do seu micro e a porta 901 do servidor, que permite acessar o Swat de forma segura.

Com o túnel ativado, você acessa o Swat usando o endereço <http://localhost:901>, como se estivesse sentado na frente do servidor. O SSH se encarrega de transportar as informações de forma transparente.

Ao abrir o Swat, você verá um menu como o do screenshot abaixo, com vários links para a documentação disponível sobre o Samba, que você pode consultar para se aprofundar no sistema. Na parte de cima, estão os links para as seções da configuração, que é o que nos interessa:



Na seção **Password**, você pode cadastrar usuários, substituindo o uso manual do comando "smbpasswd -a". Neste caso, você precisará primeiro cadastrar os usuários no sistema, utilizando o 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.

A opção "**netbios name**" indica o nome do servidor, através do qual ele será identificado na rede Windows. Normalmente se utiliza o nome da máquina, mas isso não é obrigatório, já que o nome de máquina utilizado pelo Samba não está relacionado ao nome definido no arquivo "/etc/hosts" ou à configuração do DNS. O nome pode ter até 15 caracteres e ser composto por letras e números, além de espaços e dos caracteres a seguir: ! @ # \$ % ^ & () - ' { } ~.

Ao usar mais do que 15 caracteres, os caracteres excedentes serão ignorados. É também permitido o uso de pontos, mas usá-los não é uma boa idéia, pois torna os nomes difíceis de diferenciar de nomes de domínio, o que pode confundir os usuários.

A opção "**workgroup**" indica o grupo de trabalho ao qual ele pertence. Você pode tanto utilizar o mesmo grupo de trabalho em todas as máquinas da rede, quanto agrupar suas máquinas em grupos distintos como "diretoria", "vendas", etc.



A seguir temos a opção "**interfaces**", que permite limitar os acessos ao servidor se você tiver mais de uma placa de rede. É o caso, por exemplo, de quem acessa via ADSL ou cabo e possui uma segunda placa de rede para compartilhar a conexão com os micros da rede local. Nestes casos, a placa da web será reconhecida como **eth0**, enquanto a placa da rede local será reconhecida como **eth1**, por exemplo.

Você pode, então, preencher o campo com o endereço da placa da rede local (eth1). Assim, o Samba só aceitará conexões vindas dos micros da rede local, descartando automaticamente todas as tentativas de acesso vindas da internet. Caso o campo permaneça vazio, o Samba permite acessos vindos de todas as placas de rede, e é necessário bloquear os acessos provenientes da internet usando o firewall.

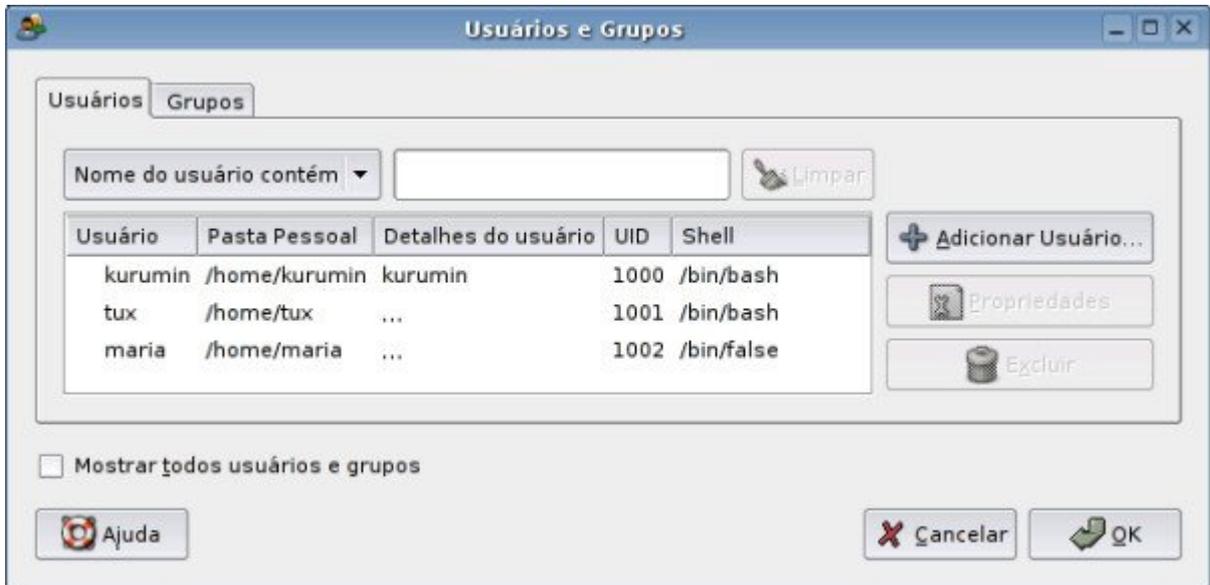
Na seção **Security Options** temos uma opção capciosa, que é a opção "security" que aceita os valores "user", "share", "server" e "domain". Com nomes tão descritivos a configuração fica fácil, já que "server" é para quando estamos configurando o Samba como servidor e "domain" é para quando ele está sendo configurado como controlador de domínio, certo? Errado! :)

As opções share e server são opções obsoletas (como veremos em detalhes mais a seguir) e a opção "domain" é usada quando você deseja que o servidor Samba seja configurado como membro (cliente) de um domínio sob responsabilidade de outro servidor. Se você está configurando um servidor Samba, seja como um servidor de grupo de trabalho, seja como controlador de domínio, a opção correta para a opção security é a "**user**".

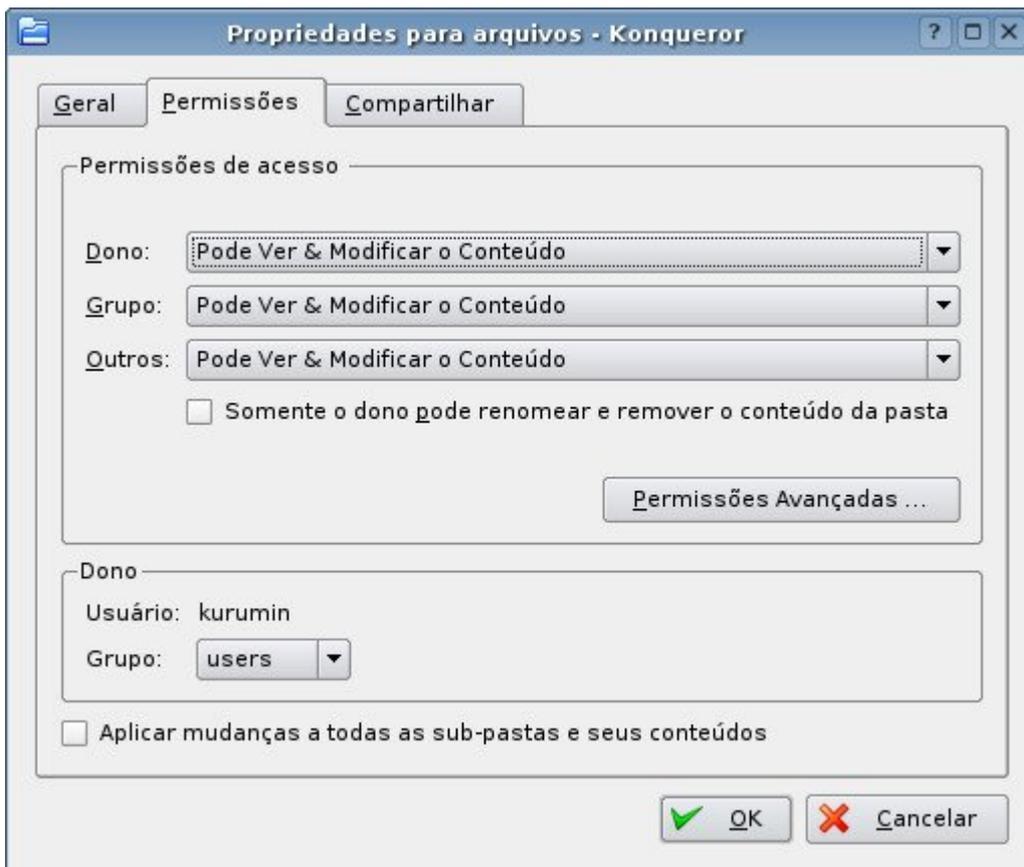
Utilizando o modo user, as permissões de acesso aos compartilhamentos do samba ficam condicionadas às permissões de acesso de cada usuário. Por exemplo, se você compartilhar a pasta **/home/maria/arquivos**, por default apenas a usuária maria terá permissão para gravar novos arquivos e alterar o conteúdo 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 mesmo. Outra opção

é adicionar os demais usuários no grupo "maria" (cada usuário possui um grupo com o mesmo nome do login, criado no momento em que é cadastrado) e configurar as permissões de acesso de forma que o grupo possa escrever na pasta. Você pode fazer a administração de grupos usando o "**users-admin**", que facilita bastante as coisas ao trabalhar com um grande número de usuários. Lembre-se que no Debian ele é instalado através do pacote "gnome-system-tools". No Fedora ele se chama "system-config-users".



Se você não está tão preocupado com a segurança, pode fazer do jeito "fácil", alterando a opção "outros" nas permissões de acesso da pasta, que dá acesso a todo mundo. Isso faz com que qualquer usuário local do sistema (ou logado via SSH) tenha acesso aos arquivos da pasta, mas não permite necessariamente que outros usuários do Samba possam acessar, pois neste caso ainda são usadas as permissões de acesso no Samba. A alteração das permissões da pasta é feita usando o Konqueror ou outro gerenciador de arquivos e não através do Samba.



Ou seja, é necessário fazer com que os usuários do grupo, ou todos os usuários do sistema, possam escrever na pasta, evitando que as permissões do sistema conflitem com as permissões configuradas no Samba. Se configuro o Samba para permitir que o usuário "joao" possa escrever no compartilhamento, mas a configuração das permissões da pasta compartilhada não permitem isso, o joao vai continuar sem conseguir escrever. Ao criar compartilhamentos no Samba, é preciso se preocupar com as duas coisas.

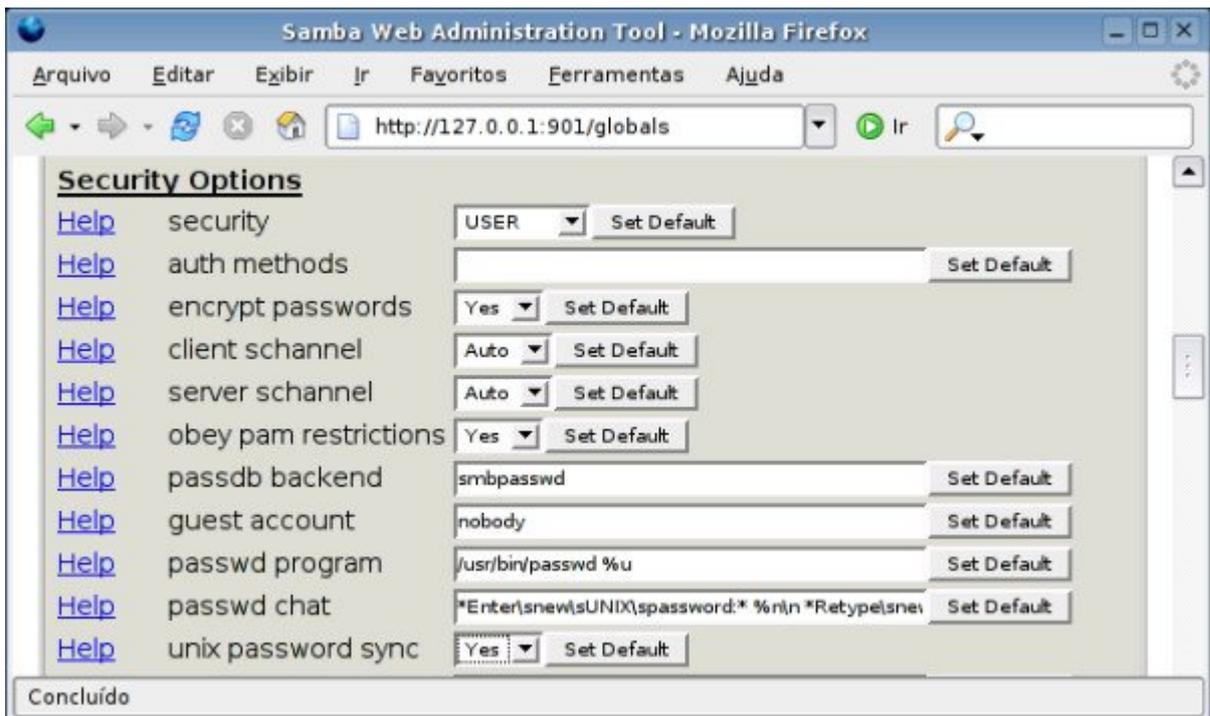
Mais abaixo, temos a opção **Encrypt Password**. Ela também é importante, e deve ficar sempre ativada (Encrypt Password = yes).

Todas as versões do Windows, incluindo o 3.11 suportam o uso de senhas encriptadas, mas até o Windows 95 original os clientes deixavam de usar a encriptação e passavam a enviar as senhas em texto puro quando percebiam que o interlocutor não suportava encriptação. Entretanto, isso abria margem para todo tipo de ataques, de forma que a partir do Windows 95 OSR/2 e do Windows NT 4 SP3, senhas em texto puro deixaram de ser suportadas, de forma que ao desativar o uso de senhas encriptadas no Samba, o servidor simplesmente não conseguirá conversar com as máquinas Windows e você vai ficar quebrando a cabeça até se lembrar deste parágrafo ;).

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. Antigamente, sempre que você alterava a senha de um usuário no Samba, usando o "smbpasswd", precisava alterar também a senha do sistema, usando o comando "passwd". As duas senhas precisam ficar em sincronismo, do contrário caímos no problema das permissões, onde o Samba permite que o usuário acesse o compartilhamento, mas o sistema não permite que o Samba acesse os arquivos no disco.

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 (respectivamente) os valores: "/usr/bin/passwd %u" e "*Enter\snew\sUNIX\spassword:* %n\n *Retye\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. Se quiser que todos os micros da rede tenham acesso, basta escrever apenas a primeira parte do endereço IP, como em "**192.168.0.**", onde todos os endereços dentro do escopo serão permitidos. Se for incluir mais de um endereço ou mais de um escopo de endereços, separe-os usando vírgula e espaço, como em: "192.168.0., 10.0.0., 123.73.45.167". Caso o campo permaneça vazio, a opção fica desativada e todos os micros que tiverem acesso ao servidor Samba poderão acessar.



A opção "**Hosts Deny**", por sua vez, permite especificar máquinas que não terão permissão para acessar o servidor. Estas duas opções possuem algumas peculiaridades, sobretudo quando usadas em conjunto. Veremos mais detalhes sobre o uso das duas mais adiante.

Em uma rede Windows, uma das máquinas fica sempre responsável por montar e atualizar uma lista dos compartilhamentos disponíveis e enviá-la aos demais, conforme solicitado. O host que executa esta função é chamado de "**Master Browser**".

De uma forma geral, todas as versões do Windows são capazes de atuar como Master Browser da rede e o cargo pode mudar de dono conforme as máquinas vão sendo ligadas e desligadas, mas o Samba executa o trabalho de forma muito eficiente, de forma que, a menos que você tenha outro servidor em posição hierarquicamente superior, é sempre interessante delegar esta tarefa ao servidor Samba.

O cargo de Master Browser é disputado através de uma eleição, onde os micros da rede enviam pacotes de broadcast contendo informações sobre o sistema operacional usado, o tempo de uptime e outras informações. Ao receber o pacote de broadcast de um "oponente", cada máquina compara suas credenciais com as do pacote recebido. Se suas credenciais forem inferiores, ela desiste da eleição, caso contrário responde enviando o pacote com suas próprias credencias. Este processo de eliminação continua até que sobre apenas uma máquina, que passa então a ser o Master Browser da rede, até que seja desconectado da rede, ou perca o cargo para outra máquina com credenciais superiores.

A principal credencial é o "OS Level", que nas máquinas Windows varia de acordo com a versão do sistema. As máquinas com o Windows NT Server, 2000 Server, 2003 Server ou 2008 Server possuem um OS Level de 32, as com o Windows NT Workstation, 2000 Professional ou qualquer versão doméstica do XP ou Vista possuem OS Level de 16 e as versões antigas do Windows (3.11, 95, 98, ME e SE) possuem OS Level de apenas 1.

Nos servidores Samba o valor é ajustado através da opção "**OS Level**", na seção Browse Options. Isso permite que você "trapaceie", fazendo com que o servidor Samba sempre ganhe as eleições. Para isso, configure esta opção com um valor alto, 100 por exemplo, para que ele sempre ganhe as eleições (você pode usar qualquer valor entre 0 e 255). O default dessa opção é 20, o que faz com que o servidor Samba ganhe de todas as máquinas Windows, com exceção das versões Server.

Para completar, deixe a opção "**Local Master**" e "**Preferred Master**" como "**Yes**". A opção "Local Master" faz com que o servidor Samba convoque uma nova eleição sempre que necessário (de forma a defender o cargo caso outra máquina tente assumir a posição) e a "Preferred Master" dá a ele uma leve vantagem quando confrontado com outra máquina com o mesmo OS Level:



É importante enfatizar que você nunca deve colocar dois servidores Samba na rede com o mesmo OS Level e com a opção "Preferred Master" ativada, caso contrário eles iniciarão uma disputa interminável pelo cargo, o que fará com que a navegação na rede se torne intermitente. Ao usar vários servidores Samba na rede, crie uma hierarquia, usando valores diferentes para a opção OS Level.

Se, por outro lado, você não deseja que o servidor Samba participe das eleições (caso já tenha outro servidor desempenhando este papel), basta definir a opção "Local Master" com o valor "no".

Abaixo, deixe a opção **WINS Support** ativada (**Yes**) para que o servidor Samba atue como um servidor WINS para os demais micros da rede. 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 smb.conf ao ser aberto, lendo as opções configuradas e mostrando-as na interface, mas gera um novo arquivo sempre que você clica 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. Em geral, quem tem o hábito de editar manualmente o smb.conf, acaba nunca usando o Swat e vive-versa.

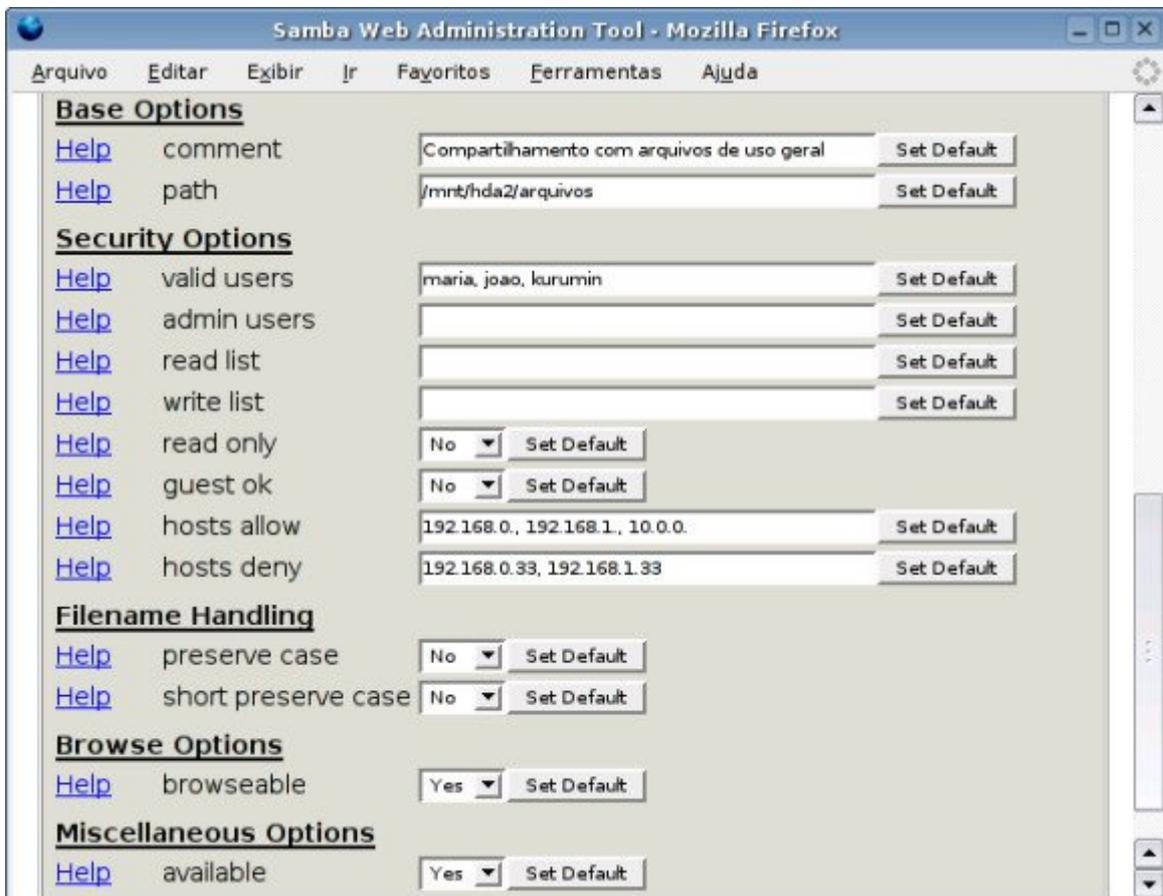
Depois de cadastrar os usuários no sistema e no Samba e configurar a seção Globals, falta apenas configurar as pastas que serão compartilhadas com as estações, através da seção "**Shares**".

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 (opção **Yes**) ou se os usuários poderão também gravar arquivos (opção **No**). Você pode também determinar quais máquinas terão acesso ao compartilhamento através das opções "**Hosts Allow**" e "**Hosts Deny**". Note que as configurações das opções "Hosts Allow" e "Hosts Deny" incluídas na seção global possuem precedência sobre as colocadas dentro da configuração dos compartilhamentos, por isso (salvo poucas exceções), elas não são usadas em conjunto.

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 "\$" em uma pasta compartilhada no Windows. 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) mostra 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 ajuste as permissões de acesso, como vimos anteriormente. No **Mandriva**, você pode instalar impressoras através do Control Center. No **Fedora** está

disponível o "**system-config-printer**", que contém basicamente as mesmas funções. Em outras distribuições, você pode usar o **kaddprinterwizard** ou a própria interface de administração do CUPS, que você acessa (via navegador) através da URL:

http://127.0.0.1:631 (veja mais detalhes sobre o compartilhamento de impressoras na quarta parte deste tutorial).

Permitindo que os usuários compartilhem pastas

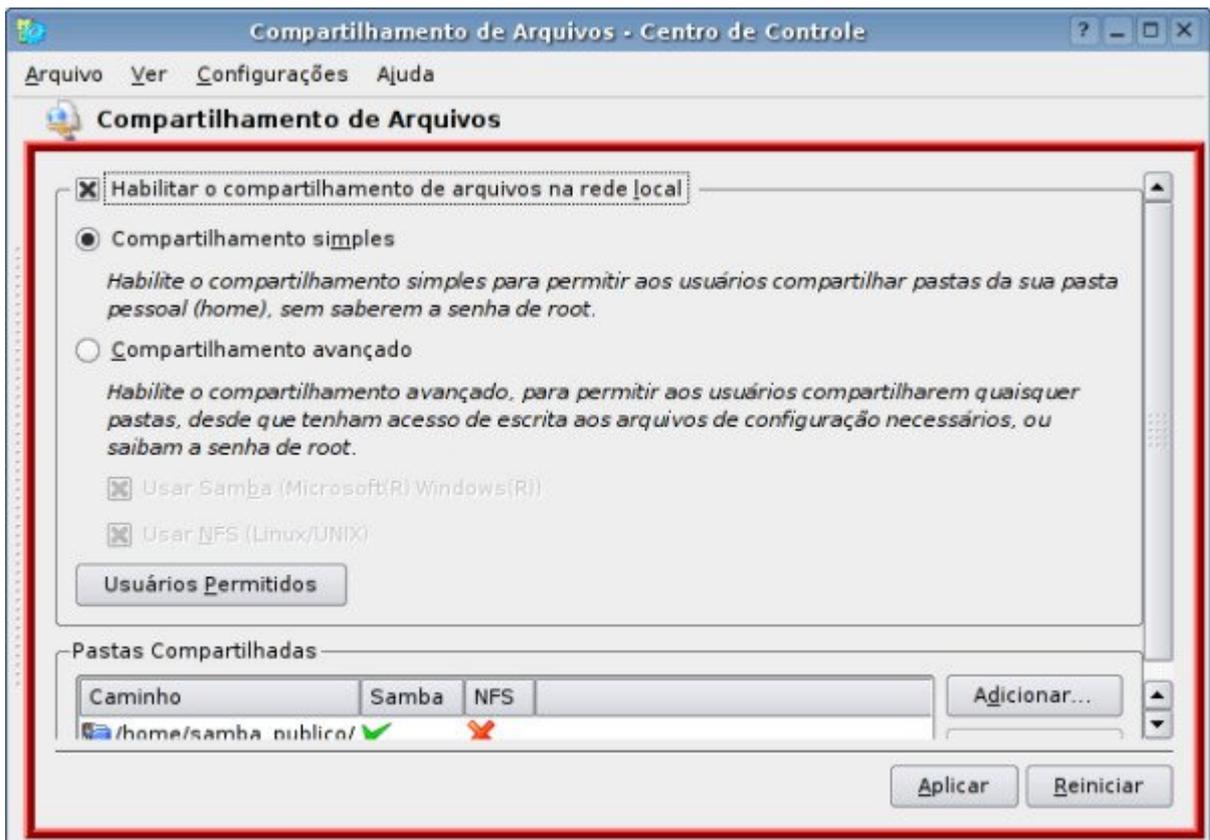
A configuração do Samba através do Swat é bem simples para configurar um servidor de arquivos, por exemplo, mas, e se você quiser permitir que os usuários também criem compartilhamentos em suas estações de trabalho Linux, assim como no Windows? Não seria muito prático ter que ensiná-los a usar o Swat ou a editarem manualmente o arquivo `smb.conf`.

O KDE possui um módulo que resolve este último problema, permitindo que os usuários compartilhem arquivos dentro dos seus respectivos diretórios de usuário de uma forma bastante simples, algo parecido com o que temos no Windows. 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**", que pode ser instalado pelo `apt-get`. Em outras distribuições ele é incluído diretamente no pacote "**kdenetwork**", que precisa estar instalado.

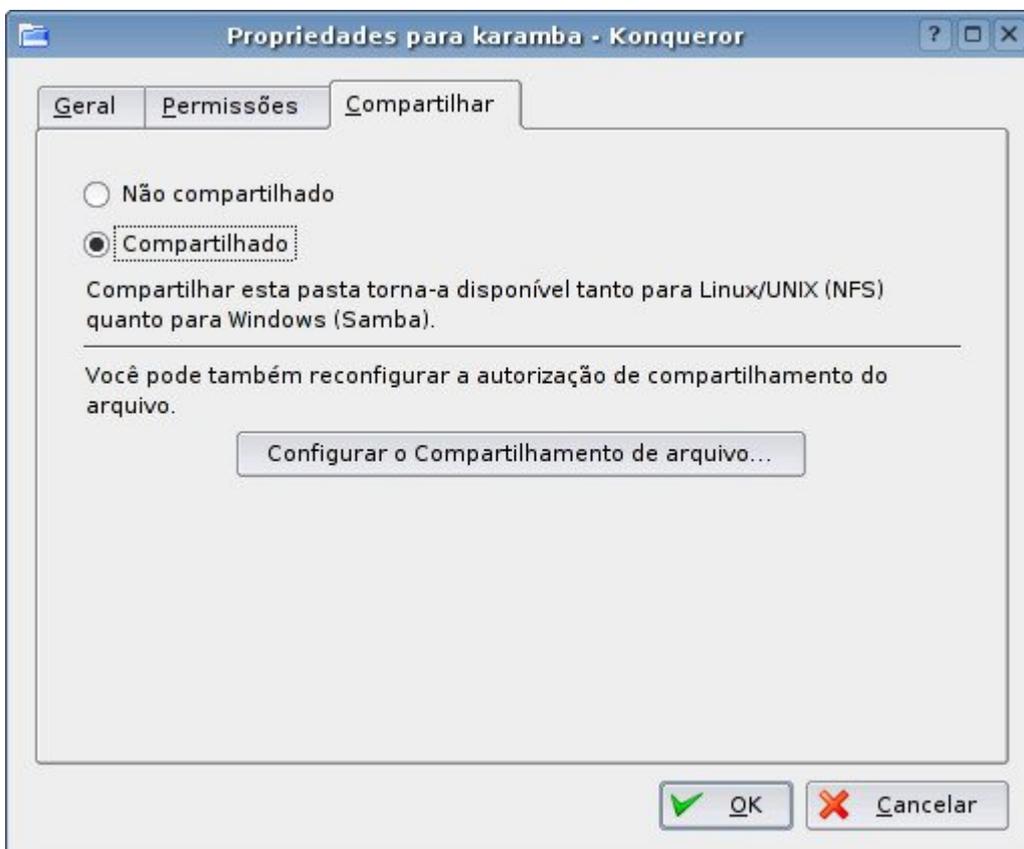
Como os usuários podem apenas compartilhar seus próprios arquivos, a possibilidade de danos ao sistema é pequena. Se você tiver um firewall isolando a sua rede local da internet, você poderá conviver com isso sem muitos sustos. :)

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 (habilite o compartilhamento simples, para permitir que os usuários compartilhem pastas de sua pasta pessoal (home), sem saberem a senha de root.)".

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:



A partir daí os usuários poderão compartilhar pastas simplesmente acessando a aba "Compartilhar", dentro das propriedades de cada uma:



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 (ou desinstalar) o serviço "nfs-kernel-server" (ou "nfs", nas distribuições derivadas do Red Hat). Naturalmente, para que o compartilhamento funcione, você deverá ter o servidor e o cliente Samba instalados no sistema e manter o serviço SMB ativo.

Samba, parte 2: Configuração avançada do Samba

Como vimos na primeira parte do tutorial, a maior parte da configuração do Samba, incluindo as configurações gerais do servidor, impressoras e todos os compartilhamentos, é feita em um único arquivo de configuração, o `/etc/samba/smb.conf`. Programas de configuração, como o Swat, simplesmente lêem este arquivo, "absorvem" as configurações atuais e depois geram o arquivo novamente com as alterações feitas. Isso permite que o Swat coexista com a edição manual do arquivo. Uma particularidade é que o Swat remove todos os seus comentários e formatação (deixando apenas as opções), por isso muitos evitam usá-lo.

Apesar disso, o formato do arquivo é bastante simples e por isso muitas vezes é mais rápido e até mais simples editar diretamente o arquivo do que através do Swat. Ao instalar o Samba, é criado um arquivo de configuração de exemplo, com vários comentários. Assim como no caso do Squid, ele é longo e difícil de entender, por isso acaba sendo mais fácil renomeá-lo e começar com um arquivo em branco, ou usar como base a configuração gerada através do Swat.

Vamos então a uma segunda rodada de explicações sobre a configuração do Samba, agora editando diretamente o arquivo `smb.conf` e explorando com maior profundidade as opções disponíveis. Vamos começar com um exemplo simplista, onde temos um único compartilhamento de teste:

```
[global]
netbios name = Sparta
workgroup = Grupo
```

```
[arquivos]
path = /mnt/arquivos
comment = Teste
```

Como você pode ver, o arquivo é dividido em seções. A primeira sempre a seção `[global]`, que contém as opções gerais do servidor. Por enquanto definimos apenas o nome do servidor (`netbios name`) e o nome do grupo de trabalho (`workgroup`), que seria o mínimo necessário para colocar o servidor na rede. As demais opções (não especificadas no arquivo) são configuradas usando os valores default. Se você omitir a opção `"workgroup"`, por exemplo, o Samba vai reverter para o grupo `"WORKGROUP"`, que é o padrão.

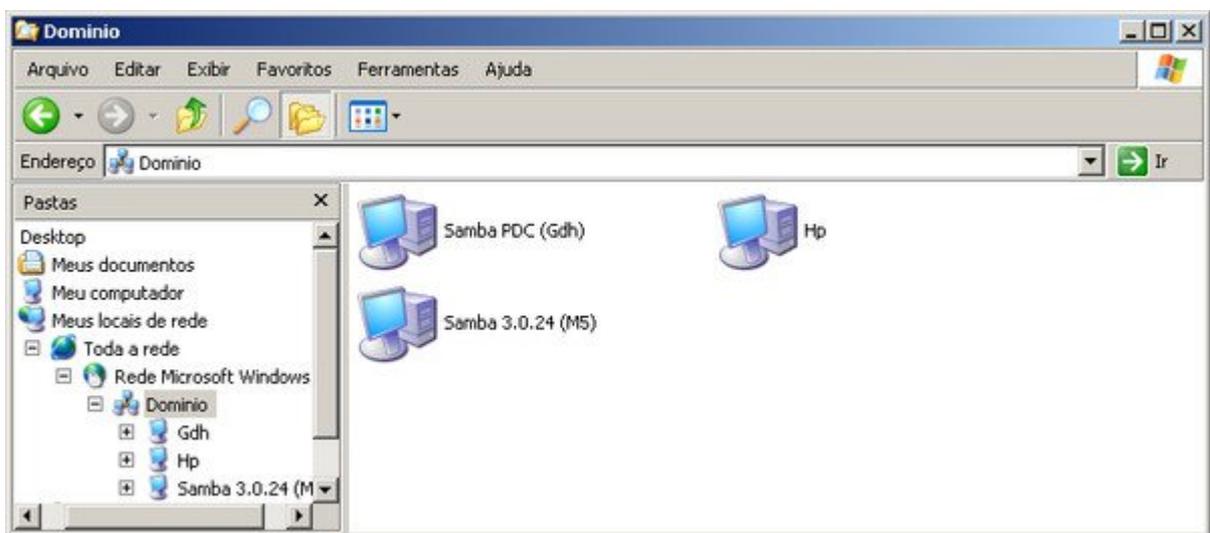
Se quiser, você pode também adicionar uma descrição para o servidor, o que é feito através da opção "server string" (adicionada dentro da seção [global]), como em:

```
server string = Servidor Samba
```

Duas dicas são que:

a) O default do Samba é usar a string "Samba 3.0.24" (onde o "3.0.24" é a versão usada) como descrição quando a opção "server string" não está presente no arquivo.

b) Nas máquinas com o Windows XP, a descrição do servidor aparece antes do nome propriamente dito, como em "Servidor Samba (Sparta)". É importante levar isso em consideração, já que no final das contas, o que importa é o que os usuários irão ver ao navegar pelo ambiente de redes:



Abaixo da seção [global], adicionamos seções adicionais para cada compartilhamento, que é o caso da seção "[arquivos]" que cria o compartilhamento de teste.

O "[arquivos]" indica o nome do compartilhamento, da forma como ele aparecerá na rede. logo a seguir temos a linha "**path**", que diz qual pasta do servidor será compartilhada e a linha "**comment**" (opcional), que permite que você inclua um comentário.

Sempre que alterar manualmente **smb.conf**, ou mesmo alterar algumas opções pelo Swat e quiser verificar se as configurações estão corretas, rode o comando **testparm**:

```
# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[arquivos]"
Loaded services file OK.
Server role: ROLE_STANDALONE
```

Ele funciona como uma espécie de debug, indicando erros grosseiros no arquivo e informando o papel do servidor na rede. O "ROLE_STANDALONE" significa que o servidor foi configurado como um membro normal do grupo de trabalho. É possível

também fazer com que o servidor Samba atue como um controlador de domínio, como veremos em detalhes na terceira parte deste tutorial.

Em caso de erros no arquivo, o testparm ajuda a localizar o problema, indicando a linha ou opção inválida, de forma que você possa corrigí-la. Veja o que acontece ao adicionar um erro simples, usando a linha "writable = yes" no lugar de "writtable = yes":

```
Unknown parameter encountered: "writable"  
Ignoring unknown parameter "writable"
```

O Samba não diferencia o uso de maiúsculas e minúsculas nas opções, de forma que tanto faz escrever "writable = yes", "writable = Yes" ou "writable = YES". Entretanto, muitos dos parâmetros não são diretamente usados pelo Samba, mas sim repassados ao sistema que, diferentemente do Samba, diferencia os caracteres. Um exemplo são as localizações de pastas a compartilhar. Se você escrever "path = /mnt/Arquivos" (em vez de "path = /mnt/arquivos"), o compartilhamento não vai funcionar, pois o sistema reportará que a pasta não existe.

Além do caractere "#", é possível usar também o ";" para comentar linhas. A principal observação é que você não pode inserir comentários em linhas válidas (mesmo que no final da linha), pois ao fazer isso toda a linha passa a ser ignorada pelo Samba. Neste exemplo, o comentário foi incluído na linha "path", o que acaba por desativar o compartilhamento completamente:

```
[teste]  
path = /mnt/arquivos # Pasta compartilhada  
comment = Compartilhamento que não funciona
```

O testparm também não indica o erro diretamente, já que ele também considera a linha como um comentário, o que pode levá-lo a perder uma boa dose de tempo tentando descobrir onde está o problema. Ao incluir comentários no arquivo, use sempre linhas separadas.

```
[teste]  
# Pasta compartilhada  
path = /mnt/arquivos  
comment = Agora sim
```

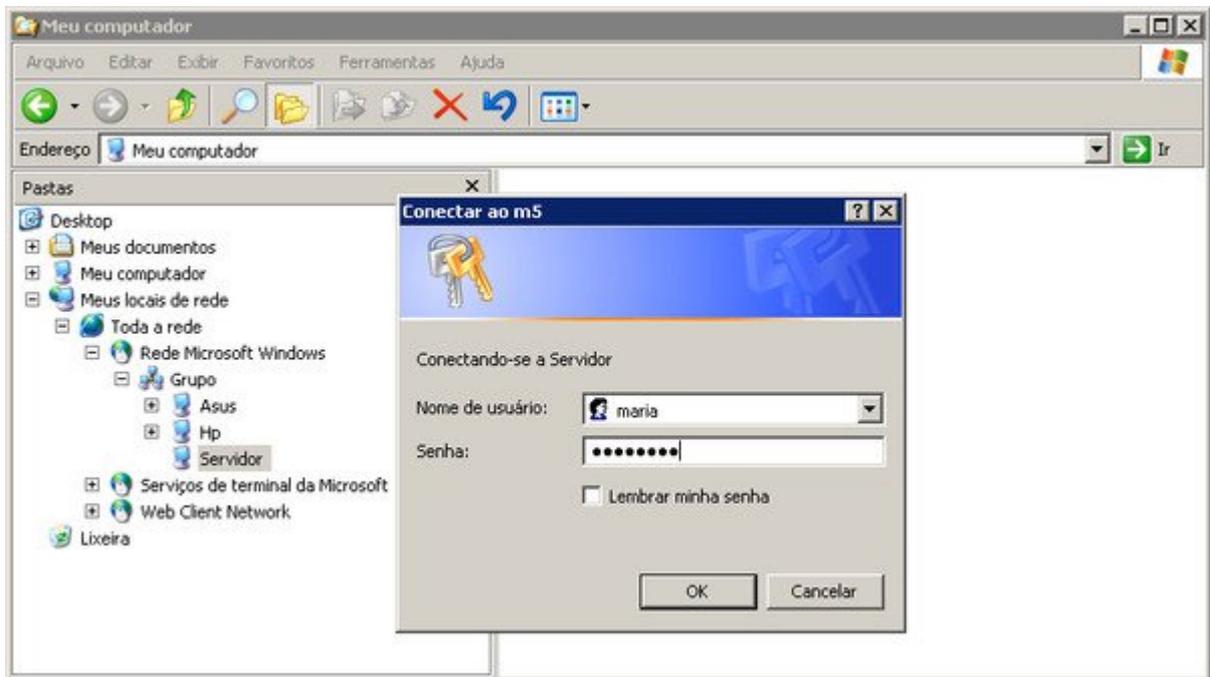
As alterações no arquivo são lidas periodicamente pelo Samba (o default são 3 minutos) e aplicadas automaticamente. Isso permite que as mudanças de configuração sejam aplicadas de forma suave, sem prejudicar o acesso dos usuários, o que é importante em um ambiente de produção. Para fazer com que as alterações entrem em vigor automaticamente, reinicie o serviço do Samba:

```
# /etc/init.d/samba restart
```

ou:

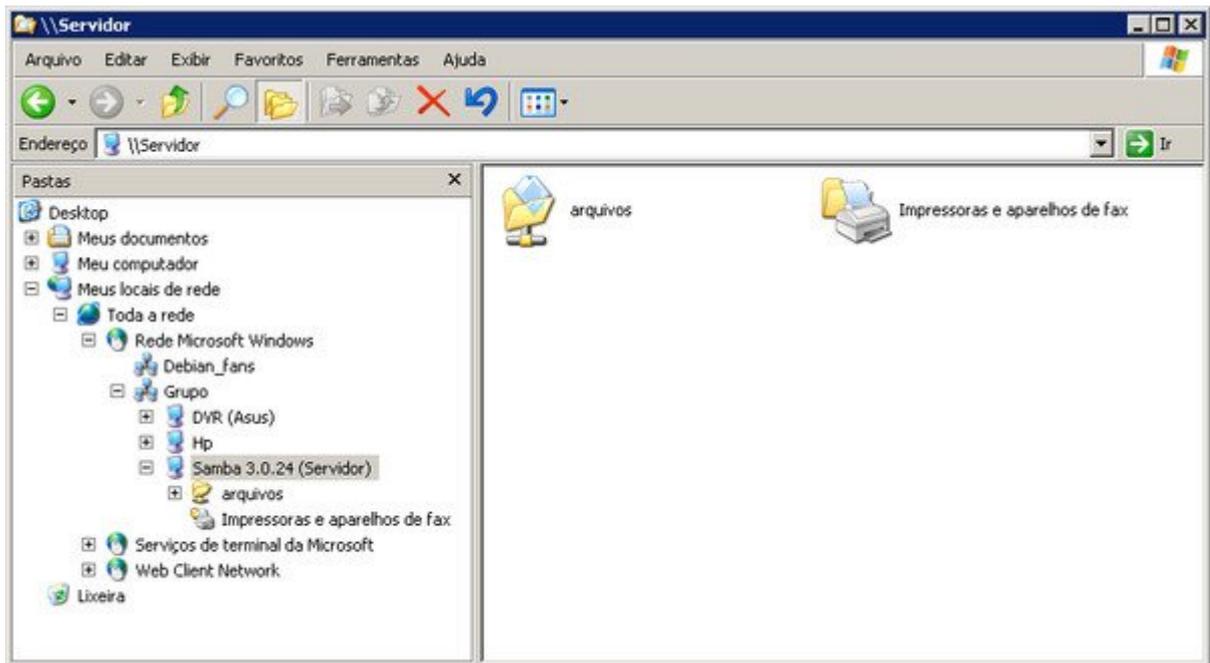
```
# /etc/init.d/smb restart
```

A partir daí o compartilhamento estará disponível. Ao tentar acessar o servidor através do "Meus locais de rede" nos clientes Windows, você receberá um prompt de senha, onde você precisa fornecer um dos logins cadastrados no servidor usando o comando "smbpasswd -a":



É importante enfatizar que todos os usuários cadastrados no Samba precisam também existir no sistema, por isso, antes de usar o comando "smbpasswd -a", você deve usar o adduser para criar o usuário. Como citei anteriormente, a solução para casos em que você não deseja criar contas válidas para todos os usuários é criar usuários limitados usando o comando "adduser --disabled-login --no-create-home usuario" ou "adduser -M usuario".

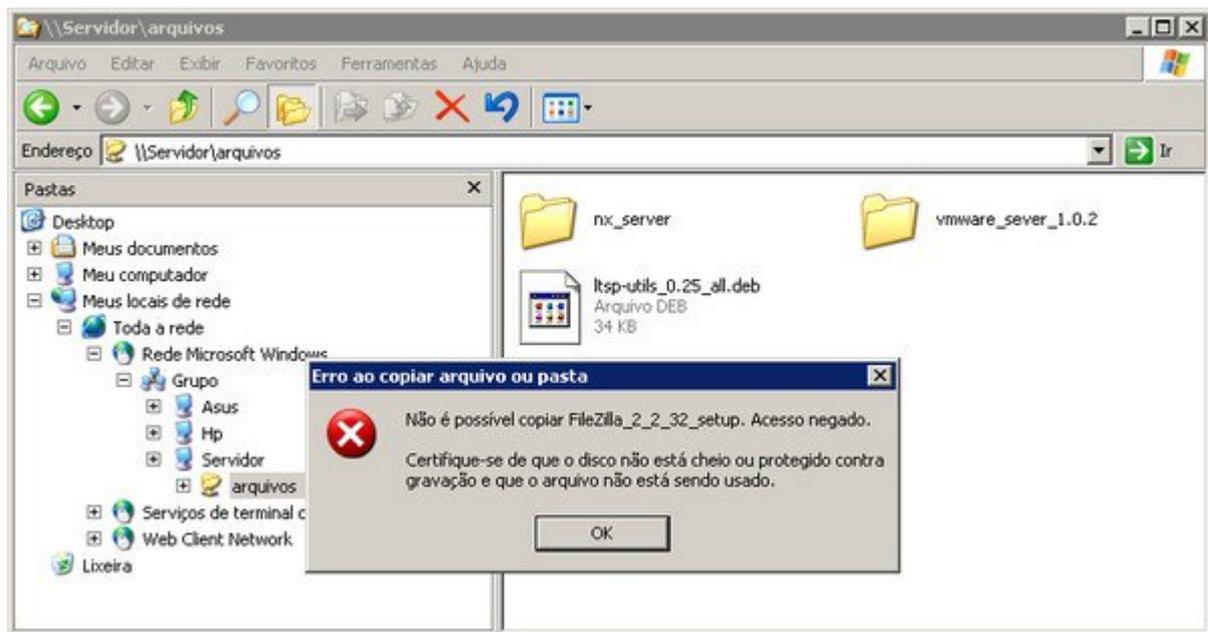
Depois de logado, o cliente pode visualizar os compartilhamentos do servidor. Por enquanto tempo apenas o compartilhamento "arquivos", que mostra o conteúdo da pasta "/mnt/arquivos" do servidor:



Ajustando as permissões de acesso

Com esta configuração, os clientes conseguem visualizar os arquivos da pasta normalmente, mas ainda não conseguem gravar nos arquivos:





Isso acontece por dois motivos. A primeiro é que o default do Samba é compartilhar com permissão apenas para leitura. Como não dissemos nada sobre as permissões de acesso do compartilhamento no arquivo de configuração, ele foi compartilhado usando o default.

Para que o compartilhamento fique disponível com permissão de leitura e escrita, precisamos adicionar a opção "writable = yes" dentro da configuração do compartilhamento, que ficará:

```
[arquivos]
path = /mnt/arquivos
writable = yes
comment = Teste
```

Muito provavelmente, mesmo depois de reiniciar o Samba você continuará recebendo o mesmo erro ao tentar gravar os arquivos. Dessa vez, o Samba autoriza a gravação, mas ela ainda pode ser abortada se as permissões da pasta não permitam que o usuário grave arquivos. Se você criou a pasta "/mnt/arquivos" como root, então o default é que apenas ele possa gravar arquivos na pasta. Para permitir que outros usuários possam gravar, é necessário abrir as permissões da pasta.

A esta altura, a lei do mínimo esforço diria para você usar um:

```
# chmod 777 /mnt/arquivos
```

Obviamente, isso permitiria que os usuários gravassem na pasta. O problema é que as permissões ficariam escancaradas, a ponto de qualquer um, que tenha acesso ao servidor (por qualquer meio) possa alterar os arquivos dentro da pasta, o que não é nada bom do ponto de vista da segurança.

No tópico sobre o Swat vimos como criar um grupo usando o "system-config-users" e abrir as permissões da pasta apenas para os usuários que fazem parte dele. Vamos ver agora como fazer isso via linha de comando.

O primeiro passo é criar um grupo para os usuários que poderia fazer alterações na pasta, usando o comando "addgroup". Eu prefiro criar grupos com o mesmo nome do compartilhamento, para ficar mais fácil de lembrar, mas isso fica a seu critério.

```
# addgroup arquivos
```

A partir daí, você pode adicionar usuários ao grupo usando o comando "adduser", nesse caso especificando o usuário já criado e o grupo ao qual ele será adicionado, como em:

```
# adduser joao arquivos  
# adduser maria arquivos
```

Para remover usuários do grupo, você usa o comando "deluser", como em:

```
# deluser joao arquivos  
# deluser maria arquivos
```

Depois de criar o grupo de adicionar os usuários a ele, falta apenas ajustar as permissões de acesso da pasta, de forma que o grupo tenha acesso completo, como em:

```
# chgrp arquivos /mnt/arquivos  
# chmod 775 /mnt/arquivos
```

Com isso, trocamos o grupo dono da pasta e dizemos que tanto o dono quanto o grupo possuem acesso completo. A partir desse ponto, o Samba autoriza o acesso para todos os usuários cadastrados através do smbpasswd e o sistema autoriza a gravação para todos os usuários que fazem parte do grupo.

Se você precisar que a alteração seja aplicada de forma recursiva, alterando as permissões de todas as subpastas e arquivos, adicione a opção "-R" nos dois comandos, como em:

```
# chgrp -R arquivos /mnt/arquivos  
# chmod -R 775 /mnt/arquivos
```

Além de servirem para controlar as permissões de acesso dos usuários às pastas do sistema, os grupos podem ser usados para ajustar as permissões de acesso do Samba, de forma bastante simples.

Outra configuração bastante usada é ativar o stick bit, um parâmetro adicional que faz com que cada usuário possa apagar apenas os seus próprios arquivos (mesmo que as demais permissões digam o contrário), o que adiciona uma camada de segurança ao abrir as permissões de acesso a uma determinada pasta.

Para ativar o stick bit, adicione um "1" antes dos 3 números com as permissões ao rodar o chmod, como em: **chmod 1777 /mnt/arquivos**

Se você quer que o compartilhamento fique disponível apenas para os usuários que cadastrou no grupo "arquivos", adicione a opção "valid users = +arquivos" na seção referente ao compartilhamento. O "+" indica que se trata de um grupo e não de um usuário isolado. O Samba verifica então quais usuários fazem parte do grupo e autoriza o acesso. A partir daí, quando você quiser liberar o acesso para um novo usuário, basta adicioná-lo ao grupo.

```
[arquivos]
path = /mnt/arquivos
writable = yes
valid users = +arquivos
```

Você pode também especificar uma lista de usuários isolados, separando-os por vírgula, por espaço, ou pelos dois combinados (o que preferir), como em:

```
[arquivos]
path = /mnt/arquivos
writable = yes
valid users = joao, maria, jose
```

É possível também combinar as duas coisas, indicando um ou mais grupos e também alguns usuários avulsos, como em:

```
[arquivos]
path = /mnt/arquivos
writable = yes
valid users = +arquivos, jose, joaquim, +admin
```

Assim como na maioria das opções do Samba, a opção "valid users" é exclusiva, ou seja, ao dizer que os usuários do grupo arquivos devem ter acesso, você automaticamente exclui todos os outros. Você pode também fazer o oposto, criando uma lista de usuários que não devem ter acesso e mantendo o acesso para os demais. Nesse caso você usaria a opção "invalid users", como em:

```
[arquivos]
path = /mnt/arquivos
writable = yes
invalid users = jose, joaquim
```

Nesse caso, todos os usuários cadastrados no Samba podem acessar, com exceção dos usuários jose e joaquim. É possível ainda usar a opção "invalid users" para especificar exceções ao especificar grupos usando a opção "valid users", como em:

```
[arquivos]
path = /mnt/arquivos
writable = yes
valid users = +arquivos
invalid users = joao
```

Nesse caso, todos os usuários dentro do grupo arquivos terão acesso, com exceção do joao. Esta combinação pode ser usada em casos onde o grupo é especificado também em

outros compartilhamentos e você precisa bloquear o acesso do usuário a um compartilhamento específico, sem removê-lo do grupo.

É possível também criar uma lista de escrita, usando a opção "write list". Ela cria uma camada adicional de proteção, permitindo que, dentro do grupo de usuários com acesso ao compartilhamento, apenas alguns tenham permissão para alterar os arquivos, como em:

```
[arquivos]
path = /mnt/arquivos
writable = no
valid users = +arquivos
write list = maria
```

Nesse caso, usamos a opção "writable = no", o que faz com que o compartilhamento passa a ser somente-leitura. A seguir, especificamos que os usuários do grupo "arquivos" devem ter acesso (somente-leitura) e usamos a opção "write list = maria" para criar uma exceção, dizendo que a maria pode escrever na pasta. É importante notar que neste exemplo a maria deve fazer parte do grupo "arquivos", caso contrário teríamos uma situação interessante, onde ela não consegue alterar os arquivos no compartilhamento pois não tem acesso a ele em primeiro lugar :).

Caso a maria não estivesse cadastrada no grupo, você deveria incluir o login na opção "valid users", como em:

```
[arquivos]
path = /mnt/arquivos
writable = no
valid users = +arquivos, maria
write list = maria
```

Podemos também fazer o oposto, restringindo a escrita para alguns usuários, mas mantendo o acesso para todos os demais. Nesse caso usamos a opção "read list" para criar uma lista de exceções, como em:

```
[arquivos]
path = /mnt/arquivos
writable = yes
valid users = +arquivos, +admin
read list = maria, jose
```

Nesse exemplo, usamos a opção "writable = yes" e especificamos que os usuários dentro dos grupos arquivos e admin tem acesso ao compartilhamento. Em seguida, usamos a opção "read list" para limitar o acesso dos usuários maria e jose, de forma que eles possam apenas ler, sem alterar os arquivos dentro da pasta.

Outra opção relacionada é a "read only", que também aceita os valores "yes" e "no". Na verdade, ela tem a mesma função da opção "writable", apenas usa uma lógica invertida. Dizer "writable = yes" ou dizer "read only = no" tem exatamente o mesmo efeito, como seis e meia-dúzia. Em geral você usa uma ou outra de acordo com o contexto, como uma forma de tornar o arquivo mais legível, como em:

```
[modelos]
path = /mnt/modelos
read only = yes
```

Continuando, é possível restringir o acesso também com base no endereço IP ou nome da máquina a partir da qual o usuário está tentando acessar o compartilhamento. Isso permite adicionar uma camada extra de segurança no acesso a arquivos importantes, já que além do login e senha, é verificado a partir de qual máquina o acesso é proveniente.

Isso é feito através das opções "hosts allow" e "hosts deny" que permite, respectivamente, criar uma lista de máquinas que podem e que não podem acessar o compartilhamento. As listas podem incluir tanto os endereços IP quanto os nomes das máquinas.

Para restringir o acesso ao compartilhamento a apenas duas máquinas específicas, você usaria:

```
[arquivos]
path = /mnt/arquivos
writable = yes
hosts allow = 192.168.1.23, 192.168.1.24
```

ou

```
[arquivos]
path = /mnt/arquivos
writable = yes
hosts allow = esparta, athenas
```

É possível também fazer o inverso, bloqueando o compartilhamento para acessos provenientes das duas máquinas. Nesse caso, mesmo que o usuário tente acessar usando um login válido, vai receber a mensagem de acesso negado, como se o login tivesse sido bloqueado ou a senha tenha sido alterada. A lista não possui um tamanho máximo, você pode incluir quantas máquinas precisar, separando os endereços ou nomes por vírgula e espaço. Você pode inclusive misturar endereços IP com nomes de máquinas:

```
[arquivos]
path = /mnt/arquivos
writable = yes
hosts deny = 192.168.1.23, athenas
```

É possível ainda combinar a restrição com base nos nomes e endereços, com a restrição com base nos logins de acesso, de forma que o acesso seja autorizado apenas quando baterem as duas coisas.

Para permitir que apenas a maria e o joao acessem o compartilhamento e ainda assim apenas se estiverem usando uma das duas máquinas permitidas, você usaria:

```
[arquivos]
path = /home/arquivos
writable = yes
```

valid users = maria, joao
hosts allow = 192.168.1.23, 192.168.1.24

Você pode autorizar ou restringir o acesso para uma faixa inteira de endereços omitindo o último octeto do endereço. Por exemplo, para que apenas clientes dentro da rede "192.168.1.x" tenham acesso, você inclui apenas a parte do endereço referente à rede, omitindo o octeto referente ao host, como em:

```
[arquivos]
path = /mnt/arquivos
writable = yes
hosts allow = 192.168.1.
```

Se precisar criar exceções, limitando o acesso a algumas máquinas dentro da faixa de endereços especificada, você pode usar a opção "EXCEPT" para especificar as exceções, como em:

```
[arquivos]
path = /mnt/arquivos
writable = yes
hosts allow = 192.168.1. EXCEPT 192.168.1.23, 192.168.1.24
```

O mesmo pode ser feito ao usar a opção "hosts deny", como em:

```
[restrito]
path = /mnt/sda2/restrito
writable = yes
valid users = isac
hosts deny = 192.168.1. EXCEPT 192.168.1.23
```

Outro parâmetro que pode ser usado ao criar exceções é o "ALL", que inclui todos os endereços possíveis. Se a idéia é que apenas um determinado endereço possa acessar o compartilhamento, uma opção é usar "hosts deny = ALL EXCEPT 192.168.1.34".

O default do Samba é permitir o acesso a partir de qualquer máquina, de forma que se você não usar nem a opção "hosts allow", nem a "hosts deny", qualquer máquina poderá acessar o compartilhamento.

Ao usar apenas a opção "hosts allow", apenas as máquinas listadas terão acesso ao compartilhamento, as demais serão recusadas. Ao usar apenas a opção "hosts deny", apenas as máquinas listadas não terão acesso ao compartilhamento (as demais continuam acessando).

Ao combinar o uso das opções "hosts allow" e "hosts deny", a opção "hosts allow" tem precedência (não importa a ordem em que elas sejam colocadas), de forma que as máquinas listadas terão acesso, mesmo que ele seja negado pela opção "hosts deny". Por exemplo, ao usar:

```
[isos]
path = /mnt/isos
hosts allow = 192.168.1.
```

```
hosts deny = 192.168.1.43
comment = Algo está errado
```

... o micro "192.168.1.43" continuará tendo acesso ao compartilhamento, pois faz parte da faixa de endereços cujo acesso é autorizado pela opção "hosts allow". Neste caso, o Samba não considera a opção "hosts deny = 192.168.1.43" como uma exceção mas sim como um erro de configuração. Para bloquear a máquina, você deveria usar:

```
[isos]
path = /mnt/isos
hosts allow = 192.168.1. EXCEPT 192.168.1.43
comment = Agora sim
```

Em situações onde você precisar restringir temporariamente o acesso a um determinado compartilhamento (para alguma tarefa de manutenção, por exemplo) você pode usar a opção "available = no", como em:

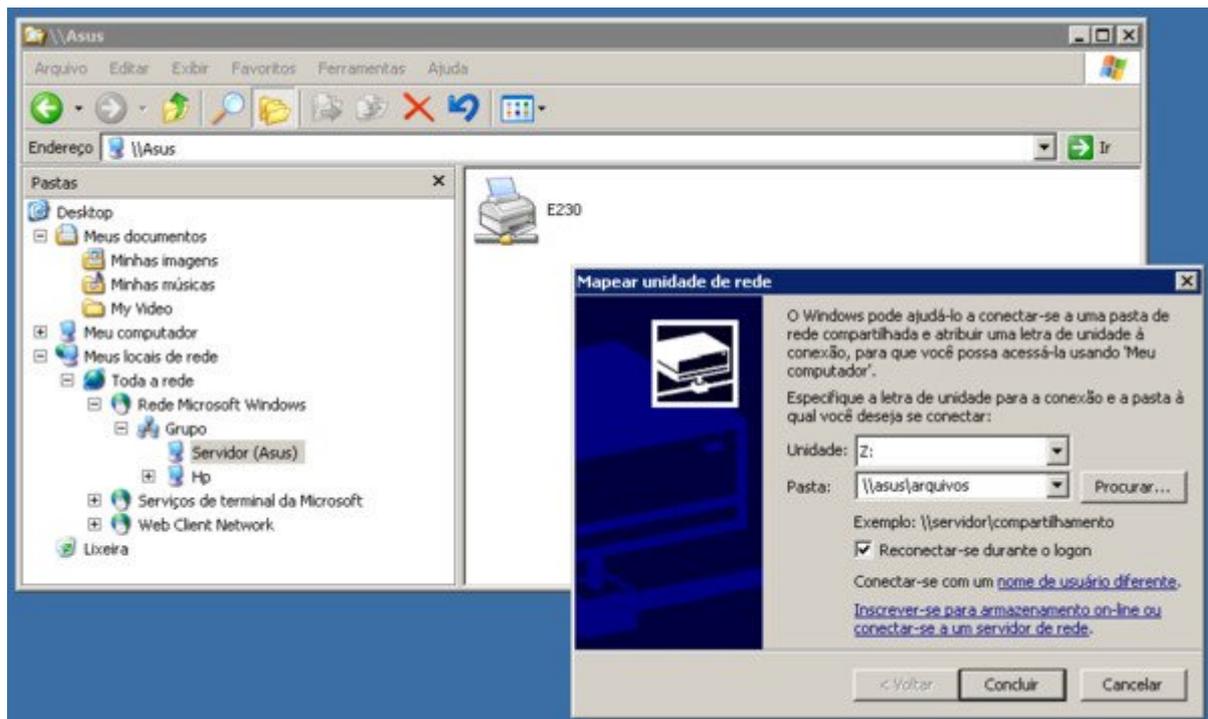
```
[arquivos]
path = /home/arquivos
writable = yes
valid users = maria, joao
available = no
```

Ela faz com que o compartilhamento "desapareça", da mesma forma que se você apagasse ou comentasse a configuração. A principal vantagem é que ao apagar você precisaria escrever tudo de novo para reativar o compartilhamento, enquanto ao usar o "available = no" você precisa apenas remover a opção ou mudar para "available = yes".

Outra opção interessante que pode ser incluída é a "browseable = no", que transforma o compartilhamento em um compartilhamento oculto:

```
[arquivos]
path = /home/arquivos
writable = yes
browseable = no
```

Com isso, ele não aparece mais no ambiente de redes, mas pode ser acessado normalmente se você especificar o nome manualmente ao mapear o compartilhamento:



Essa não é propriamente uma opção de segurança, mas pode ser usada para afastar os curiosos dos compartimentos com acesso restrito.

Concluindo, Muitas opções que ficam disponíveis no swat podem ser omitidas ao configurar manualmente, simplesmente por que são o default no Samba. O próprio Swat evita incluir opções redundantes ao gerar o arquivo, incluindo apenas as configurações que são diferentes dos valores default.

Não é necessário incluir opções como "writable =no", "available = yes" ou "browseable = yes" no arquivo, simplesmente por que estes já são os valores usados por padrão no Samba. Apesar disso, usá-los também não atrapalha em nada, de forma que nada impede que você os inclua no arquivo para se lembrar mais facilmente das opções.

Outra dica é que você pode verificar a qualquer momento quais usuários e quais máquinas estão acessando compartimentos no servidor usando o comando "smbstatus, como em:"

```
# smbstatus
```

```
Samba version 3.0.24
```

```
PID Username Group Machine
```

```
-----  
17107 gdh gdh hp (192.168.1.2)  
11588 gdh gdh semprao (192.168.1.10)
```

```
Service pid machine Connected at
```

```
-----  
IPC$ 17107 hp Sun Oct 28 15:54:04 2007  
arquivos 11588 semprao Sun Oct 28 15:23:59 2007
```

No locked files

Neste exemplo, por exemplo, podemos ver que o usuário "gdh" está logado no servidor a partir de duas máquinas diferentes, um indício de que duas pessoas estão utilizando a mesma conta.

A seção [global]

Todas as opções colocadas dentro da seção referente ao compartilhamento valem apenas para ele, o que permite que você crie diversos compartilhamentos diferentes e use um conjunto próprio de permissões para cada um. Estas mesmas opções, junto com um conjunto adicional podem ser especificadas de forma geral dentro da seção [global] do smb.conf.

Nos exemplos anteriores, especificamos apenas o nome do servidor e o grupo de trabalho na seção [global]. Isto é suficiente para o servidor participar da rede e compartilhar arquivos mas, naturalmente, existem muitas outras opções que podem ser usadas.

Em primeiro lugar, temos o nível de segurança do servidor, definido através da opção "security". O default no Samba 3 é usar o controle de acesso baseado em usuário, que é o mesmo modo de acesso usado pelas versões domésticas do Windows 2000, XP e Vista. Neste modo, você cadastra os logins e senhas no servidor, define as permissões de acesso e o servidor checa as credenciais dos clientes antes de autorizar o acesso, a configuração que vimos até aqui. Este modo é ativado adicionando a opção "**security = user**" na seção [global], mas não é necessário usá-la no Samba 3, pois, como disse, ela é usada por padrão:

```
security = user
```

Em seguida, temos o modo "security = domain". Ao contrário do que o nome pode sugerir à primeira vista, este modo não é destinado a fazer com que o Samba atue como um controlador de domínio. Pelo contrário, ao configurar um servidor Samba como PDC, você continua usando a opção "security = user", da mesma forma que faria ao usar um servidor em modo stand alone. A opção "security = domain" é usada quando você quer que um servidor Samba participe do domínio como cliente, autenticando-se em um servidor PDC já existente (que pode tanto ser outro servidor Samba, quanto ser um servidor Windows).

Existem ainda os modos "security = share" e "security = server", que imitam o sistema de acesso utilizado por estações Windows 95/98. Estes dois modos são obsoletos e devem ser removidos em futuras versões do Samba. Antigamente, o modo "security = share" era usado em casos onde você queria disponibilizar compartilhamentos públicos na rede, sem muita segurança, mas hoje em dia isso pode ser feito usando a conta guest (como veremos em detalhes mais adiante). O modo "security = server" descende da época em que o Samba ainda não era capaz de atuar como PDC; este modo permite que ele atue como um proxy de autenticação, repassando as requisições para o servidor de autenticação principal. Atualmente este modo não é mais usado.

Outra opção usada por padrão no Samba 3 é a "**encrypt passwords = yes**", de forma que também não é necessário especificá-la manualmente no arquivo. Entretanto, é saudável

incluí-la em modelos e exemplos de configuração pois pode acontecer de alguém tentar usar o modelo no Samba 2, onde o default era que ela ficasse desativada.

```
encrypt passwords = yes
```

É muito comum que seja incluída também a opção "**invalid users = root**", uma medida de segurança para evitar que a conta de root seja usada ao acessar o servidor. A lógica é que a conta de root é a única conta presente em qualquer sistema Linux, de forma que alguém que decidisse usar um ataque de força bruta para tentar obter acesso ao servidor, testando todas as senhas possíveis, começaria justamente pela conta de root. Entretanto a conta de root é necessária para dar upload de drivers de impressão e para logar os clientes ao usar o servidor Samba como PDC, situações onde ela deve ser comentada ou removida.

Continuando, as opções definidas dentro da seção [global] valem para todos os compartilhamentos. Por exemplo, ao usar:

```
[global]
netbios name = Servidor
workgroup = Grupo
hosts allow = 192.168.1.
```

... apenas as máquinas dentro da faixa de endereços especificadas terão acesso ao servidor (vale para todos os compartilhamentos), o que seria interessante do ponto de vista da segurança, já que o servidor deve ser acessado apenas por clientes da rede local de qualquer forma.

O maior problema é que a opção "hosts allow" usada na seção [global] tem precedência sobre qualquer opção "hosts deny" usada dentro dos compartilhamentos, o que pode criar problemas caso você queira restringir o acesso de alguma máquina da rede local a um determinado compartilhamento.

Por exemplo, ao usar:

```
[global]
netbios name = Servidor
workgroup = Grupo
hosts allow = 192.168.1.
```

```
[share]
path = /mnt/sda2/shared
hosts deny = 192.168.1.2
```

... a máquina "192.168.1.2" continuaria tendo acesso ao compartilhamento [share], pois o acesso é autorizado pela opção "hosts allow = 192.168.0." usada na seção [global]. Nesse caso, o melhor seria remover a linha "hosts allow = 192.168.1." da seção [global] e deixar apenas a opção "hosts deny = 192.168.1.2" na seção [share]:

```
[global]
netbios name = Servidor
workgroup = Grupo
```

```
[share]
path = /mnt/sda2/shared
hosts deny = 192.168.1.2
```

Em caso de conflito direto entre uma regra definida na seção [global] e outra definida em um dos compartimentos, a regra definida na seção [global] tem precedência. Por exemplo, ao usar:

```
[global]
netbios name = Servidor
workgroup = Grupo
hosts deny = 192.168.1.3
```

```
[share]
path = /mnt/sda2/shared
hosts allow = 192.168.1.3
```

... a máquina "192.168.1.3" continuará sem acesso ao compartilhamento "share" (ou a qualquer outro recurso do servidor), já que vale a regra definida na seção [global]. Enquanto a linha "hosts deny = 192.168.1.3" não for removida, a máquina não terá acesso a nenhum dos compartimentos, não importa o que digam as demais linhas do arquivo.

Um problema comum enfrentado ao administrar uma rede mista são os usuários escreverem a primeira letra do login em maiúsculo, como em "Joao" no lugar de "joao". No Windows isso não é um problema, já que o sistema é case insensitive, mas no Linux faz com que o sistema recuse o login. Uma forma de evitar isso no Samba é usar a opção "username level", como em:

```
username level = 2
```

Esta opção faz com que o Samba verifique várias combinações de maiúsculas e minúsculas caso o login seja recusado pelo sistema. O número indica o volume de variações, que pode ser qualquer número inteiro. Ao usar 2, o Samba verifica até dois níveis, incluindo variações como JOao, jOAO, Joao, jOaO e assim por diante. Usar um número maior pode retardar a autenticação, já que o Samba precisará testar muitas combinações, por isso são geralmente usados os valores 1 ou 2.

Outra peculiaridade digna de nota é a questão dos nomes de arquivos. No Windows, os nomes de arquivos são salvos da forma como digitados pelo usuário, preservando os caracteres maiúsculos e minúsculos. Entretanto, o sistema é case insensitive, de forma que o sistema não diferencia um arquivo chamado "Trabalho.txt" de outro chamado "trabalho.txt".

Embora o Linux seja case sensitive, o Samba tenta emular o comportamento de uma máquina Windows ao localizar arquivos. Se o cliente pede o arquivo "Trabalho.txt", quando na verdade o arquivo armazenado na pasta se chama "trabaLho.txt" o Samba vai acabar fornecendo o arquivo correto para o cliente, pois o encontrará depois de testar diversas combinações de maiúsculas e minúsculas.

No Samba 3, este recurso funciona muito bem, mas tem a desvantagem de consumir uma certa quantidade de memória do servidor. Em um pequeno servidor de rede local, isso

não faz diferença, mas em um servidor que atende um grande número de requisições, a diferença pode se tornar considerável.

Você pode simplificar as coisas orientando o Samba a salvar todos os arquivos em minúsculas. Para isso, adicione as linhas:

```
preserve case = no  
default case = lower
```

No caso de servidores com duas ou mais interfaces de rede, sobretudo no caso de servidores conectados simultaneamente à internet e à rede local, você pode especificar qual interface será usada pelo Samba através da opção "**interfaces**", que deve ser combinada com a opção "bind interfaces only = yes". Para que o servidor escute apenas a interface eth0, ignorando tentativas de conexão em outras interfaces, você usaria:

```
interfaces = eth0  
bind interfaces only = yes
```

Por default, o Samba escuta em todas as interfaces, o que (se não houver nenhum firewall ativo) pode expor seus compartilhamentos para a Internet caso você ative o Samba em uma máquina conectada diretamente à internet, como no caso de um servidor que compartilha a conexão. É recomendável usar sempre estas duas opções, como uma forma de garantir que o Samba ficará disponível apenas na interface desejada.

Outra opção interessante é a "**netbios aliases**", que 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 ou mais vezes no ambiente de rede, como se existissem várias máquinas. Em geral isso acaba confundindo mais do que ajudando, mas pode ser útil em algumas situações, quando, por exemplo, um servidor é desativado e os compartilhamentos são movidos para outro. O novo servidor pode responder pelo nome do servidor antigo, permitindo que os desavisados continuem acessando os compartilhamentos. Para usá-la, basta adicionar a opção, seguida pelos apelidos desejados, como em:

```
[global]  
netbios name = Servidor  
netbios aliases = athenas, sparta  
workgroup = Grupo
```

No tópico sobre o swat falei sobre as opções "Local Master", "OS Level" e "Preferred Master", que definem se o servidor Samba deve participar das eleições para Master Browser e com qual nível de credencial. Para que o servidor participe com OS Level 100, você adicionaria as linhas:

```
local master = yes  
os level = 100  
preferred master = yes
```

Um segundo servidor Samba na rede poderia participar com uma credencial mais baixa, de forma a assumir o cargo apenas caso o servidor principal esteja desconectado da rede. Para isso, basta usar um valor mais baixo na opção OS Level, como em:

```
local master = yes
os level = 90
preferred master = no
```

O valor da opção OS Level é absoluto, não se trata de um sorteio. Um servidor configurado com o valor "100" ganha sempre de um com o valor "99", por exemplo.

Se você omitir as três linhas, o servidor simplesmente utiliza os valores default (local master = yes, os level = 20), que fazem com que ele participe das eleições, mas utilize credenciais baixas.

A opção "**wins support = yes**" faz com que o servidor Samba passe a trabalhar como um servidor WINS (Windows Internetworking Name Server) na rede. O WINS é um protocolo auxiliar dentro das redes Microsoft, responsável pela navegação na rede e listagem dos compartilhamentos e outros recursos disponíveis, de forma similar a um servidor DNS.

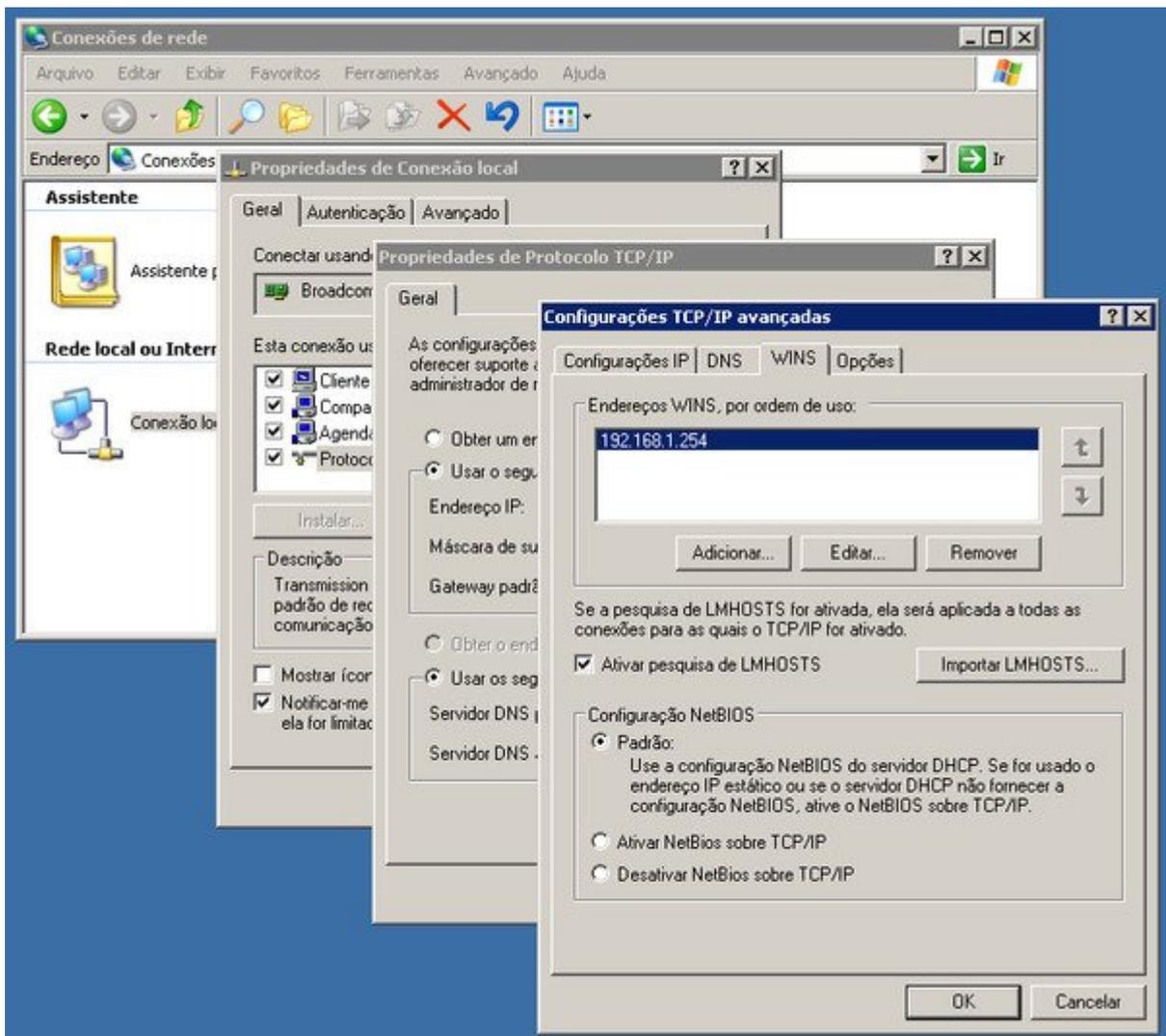
O uso do WINS não é obrigatório, sua rede vai muito provavelmente funcionar muito bem sem ele. Entretanto, sem um servidor WINS os clientes passam a usar pacotes de broadcast para a navegação, o que aumenta o tráfego da rede e torna todo o processo mais passível de falhas.

Outra limitação importante é que os pacotes de broadcast são descartados pelos roteadores, o que faz com que eles (os pacotes de broadcast) não sejam transmitidos de um segmento a outro da rede caso ela esteja dividida em vários segmentos interligados através de roteadores. O mesmo acontece caso você tenha duas redes ligadas através de algum tipo de VPN. Em ambos os casos, os pacotes de broadcast são descartados pelos roteadores, fazendo com que os micros em um segmento não enxerguem os micros do outro e vice-versa.

A solução em ambos os casos é implantar um servidor WINS na rede. Com isso, os clientes passam a consultar o servidor ao invés de mandar pacotes de broadcast, fazendo com que a navegação funcione mesmo ao utilizar vários segmentos de rede. Para isso, basta incluir a opção dentro da seção [global] do smb.conf:

```
wins support = yes
```

O próximo passo é configurar os clientes da rede para utilizarem o servidor. A opção fica escondida nas propriedades da conexão de rede, no Protocolo TCP/IP > Propriedades > Avançado > WINS, onde você deve adicionar o endereço IP do servidor:



No caso de outros micros Linux rodando o Samba que forem ser configurados como clientes do servidor principal, a configuração é feita adicionando a opção "wins server = servidor" (também na seção [global] do smb.conf), onde você especifica o endereço IP do servidor principal, como em:

```
wins server = 192.168.1.254
```

Uma observação importante é que as opções "wins support" e "wins server" são mutuamente exclusivas. Ou a máquina atua como servidor WINS, ou como cliente, nunca as duas coisas ao mesmo tempo, de forma que você não deve jamais combinar as duas opções dentro da configuração.

Em versões antigas do Samba, combinar as duas opções na configuração simplesmente faziam com que o servidor deixasse de funcionar. Nas atuais o resultado não chega a ser dramático (o servidor vai simplesmente ignorar a opção que for colocada depois), mas mesmo assim este é um erro grave de configuração que deve ser evitado.

Um exemplo de seção [global] usando as opções que vimos até aqui seria:

```
[global]
netbios name = Athenas
server string = Servidor Samba
workgroup = Grupo
username level = 1
preserve case = no
default case = lower
```

```
interfaces = eth0
bind interfaces only = yes
local master = yes
os level = 100
preferred master = yes
wins support = yes
```

A seção [homes]

Uma vantagem de utilizar usuários "reais" no servidor Samba em vez de usuários castrados é que você tem a opção de compartilhar os diretórios home através da seção [homes] no smb.conf. Este é um serviço interno do Samba, que permite compartilhar automaticamente o diretório home de cada usuário, sem precisar criar um compartilhamento separado para cada um.

A configuração mais comum é compartilhar os diretórios home com permissão de acesso apenas para o respectivo usuário. Desta forma, cada usuário tem acesso apenas ao seu próprio diretório home (que aparece no ambiente de redes como um compartilhamento com o mesmo nome), sem poder acessar, nem muito menos alterar o conteúdo dos diretórios home dos demais usuários. Nesse caso, a configuração fica:

```
[homes]
valid users = %S
read only = no
create mask = 0700
directory mask = 0700
browseable = no
```

Não é necessário especificar a pasta a compartilhar, pois ao usar o nome "homes" o Samba sabe que deve compartilhar o home de cada usuário. As opções "create mask = 0700" e "directory mask = 0700" fazem com que todos os arquivos e pastas criados pelo usuário dentro do home sejam acessíveis apenas por ele mesmo. A opção "browseable = no" faz com que cada usuário possa ver apenas seu próprio diretório, o que é reforçado pela opção "valid users = %S", que diz explicitamente que apenas o próprio usuário deve ter acesso à sua pasta home.

Uma queixa comum é que ao acessar o diretório home através do Samba, os usuários verão todos os arquivos e pastas de configuração de programas que são salvos dentro do diretório home, o que pode ser confuso.

Uma forma de evitar isso é alterar a configuração, de forma que o Samba compartilhe uma pasta vazia dentro do home, e não o diretório home em si. Com isso é mantido o propósito de oferecer uma pasta particular para o usuário, onde ele possa salvar seus

arquivos particulares e seus backups, sem a poluição gerada pela presença dos arquivos de configuração. A configuração nesse caso ficaria:

```
[homes]
path = /home/%u/share
valid users = %S
read only = no
create mask = 0700
directory mask = 0700
browseable = no
```

A linha "path = /home/%u/share" especifica que o Samba deve agora compartilhar a pasta "share" dentro do home e não mais o diretório hoje em si (você pode especificar outra pasta qualquer) e a linha "valid users = %S" garante que a pasta ficará acessível apenas para o próprio usuário.

Naturalmente, a pasta "share" precisa ser criada dentro do home de cada usuário manualmente. Você pode fazer isso de forma automática para todos os usuários usando este mini shell script:

```
cd /home
for i in *; do
mkdir $i/share
chown $i.$i $i/share
done
```

Aproveite para criar também a pasta "share" dentro do diretório "/etc/skel", que é usado como um modelo para a criação do home de novos usuários, isso faz com que o diretório seja adicionado ao home de todos os usuários criados daí em diante de forma automática:

```
# mkdir /etc/skel/share
```

Aqui vai uma lista de outras variáveis do Samba para referência:

%a : A versão do Windows usada, onde o "%a" é substituído pelas strings "Win95" (Windows 95/98), "WinNT" (Windows NT 3.x ou 4.x), "Win2K" (Windows 2000 ou XP) ou "Samba" (máquinas Linux rodando o Samba)
%I : Endereço IP da máquina cliente (ex: 192.168.1.2)
%m : Nome da máquina cliente (ex: cliente1)
%L : Nome do servidor (ex: athenas)
%u : Nome do usuário, como cadastrado no servidor Linux (ex: joao)
%U : Nome do usuário, como enviado pelo cliente Windows (pode ser diferente do login cadastrado no servidor em algumas situações)
%H : Diretório home do usuário (ex: /home/maria)
%g : Grupo primário do usuário (ex: users)
%S : Nome do compartilhamento atual (o valor informado entre colchetes, ex: arquivos)
%P : Pasta compartilhada (o valor informado na opção "path", ex: /mnt/arquivos)
%v : Versão do Samba (ex: 3.2.24)
%T : Data e horário atual

Ao longo do texto veremos alguns outros exemplos de uso destas variáveis, mas você pode usá-las em outras situações para criar compartilhamentos "inteligentes", que mostram pastas diferentes de acordo com as propriedades do cliente.

Por exemplo, a variável "%a" (que indica a versão do Windows no cliente), poderia ser usada para criar um compartilhamento com drivers, que mostrasse diretamente a pasta com os drivers corretos para a versão do Windows usada. Nesse caso, você poderia usar algo como:

```
[drivers]
path = /mnt/sda2/drivers/%a
read only = yes
```

A pasta "/mnt/sda2/drivers/" incluiria uma série de sub-pastas, com os valores possíveis para a variável, incluindo "Win95", "WinNT", "Win2K" e "Samba". Ao acessar o compartilhamento, o cliente vê apenas o conteúdo da pasta correspondente ao sistema operacional usado.

A conta guest

No Windows XP é usado por padrão um modo simplificado de compartilhamento de arquivos, o "simple sharing", que visa imitar modo de acesso do Windows 95/98, onde os compartilhamentos são públicos e você apenas define se eles são apenas leitura ou leitura e escrita.

O Windows XP usa o controle de acesso com base no usuário, assim como o Samba 3, mas, por padrão, mapeia todos os acessos para a conta guest, permitindo assim o acesso por parte de qualquer usuário da rede.

Naturalmente, podemos fazer o mesmo no Samba. Para isso, adicione as linhas abaixo dentro da seção [global] do smb.conf:

```
map to guest = bad user
guest account = guest
```

A primeira opção faz com que, sempre que um cliente especificar um usuário inválido ao tentar acessar o servidor, o servidor mapeia a requisição para o login especificado na opção "guest account", que é usada para acessar o compartilhamento. Neste exemplo, qualquer usuário não autenticado passaria a usar a conta "guest", que deve ter sido previamente cadastrada no servidor. Você pode também usar qualquer outra conta válida, como em "guest account = maria".

Uma opção menos usada é a "**map to guest = bad password**". Ela se diferencia da "map to guest = bad user" pois permite o acesso apenas caso o usuário especifique um login válido, mas erre apenas a senha. O principal motivo dela não ser muito usada é que ela confunde o usuário, já que ele ou vai achar que está realmente logado no servidor (quando na verdade está apenas acessando de forma limitada através da conta guest) ou vai passar a achar que o servidor aceita qualquer senha.

Para que os usuários não-autenticados possam acessar os compartilhamentos, você deve explicitamente autorizar o acesso, adicionando a opção "**guest ok = yes**" na configuração, como em:

```
[global]
netbios name = Sparta
workgroup = Grupo
map to guest = bad user
guest account = guest
```

```
[publico]
path = /mnt/sda2/publico
writable = yes
guest ok = yes
```

Note que no exemplo usei a opção "writable = yes". Entretanto, para que os usuários não-autenticados possam efetivamente escrever na pasta, é necessário verificar se as permissões de acesso da pasta permitem que a conta especificada ("guest" no exemplo) altere os arquivos. Como disse anteriormente, o Samba está subordinado às permissões de acesso do sistema.

Outra opção comum em compartilhamentos públicos é a "**guest only = yes**" (usada no lugar da "guest ok = yes"). Ela simula o "simple sharing" do Windows XP, mapeando qualquer acesso para a conta guest, sem sequer abrir o prompt de login para o cliente.

Vamos então a mais um exemplo de configuração do smb.conf, desta vez usando a conta guest para criar um servidor de arquivos público. Ele possui duas partições de arquivos (montadas nas pastas "/mnt/hda2 e "/mnt/sda1") que ficam disponíveis a todos os usuários da rede:

```
[global]
netbios name = Plutus
server string = Servidor público
workgroup = Grupo
local master = yes
os level = 100
preferred master = yes
wins support = yes
map to guest = bad user
guest account = gdh
```

```
[arquivos]
path = /mnt/hda2
writable = yes
guest ok = yes
```

```
[backups]
path = /mnt/sda1
writable = yes
guest ok = yes
```

Esta configuração é bastante simples e a prova de falhas. O servidor vai assumir a função de master browser, se responsabilizando pela navegação dos clientes e vai mapear qualquer acesso para a conta "gdh" usada na opção "guest account", permitindo que qualquer um possa ler e gravar arquivos nos dos compartilhamentos.

As duas principais observações são que o usuário "gdh" deve ser um usuário real do sistema e ser cadastrado no servidor Samba e ele deve ser o dono das duas pastas compartilhadas, de forma que não tenha problemas para acessar seu conteúdo. Isso pode ser feito usando os 4 comandos a seguir:

```
# adduser gdh
# smbpasswd -a gdh
# chown -R gdh.gdh /mnt/hda2
# chown -R gdh.gdh /mnt/sda1
```

Auditando os acessos

O log do Samba pode ser ativado adicionando as opções abaixo na seção [global] do smb.conf:

```
log level = 1
log file = /var/log/samba.log
max log size = 1000
```

A opção "log level" indica o nível das mensagens (e 0 a 10), sendo que o nível 0 mostra apenas mensagens críticas, o nível 1 mostra alguns detalhes sobre os acessos e os demais mostram diversos níveis de informações de debug, úteis a desenvolvedores. A opção "log file" indica o arquivo onde ele será gerado e a "max log size" indica o tamanho máximo, em kbytes.

A partir do Samba 3.04 foi incluído um módulo de auditoria, que permite logar os acessos e as modificações feitas de uma forma muito mais completa que o log tradicional. Isso é feito através do módulo "full_audit", que (do ponto de vista técnico) funciona de forma similar ao módulo "recycle" usado pela lixeira.

O primeiro passo é ativar o módulo, o que é feito através da linha abaixo:

```
vfs objects = full_audit
```

O próximo passo é definir quais operações devem ser logadas através da opção "full_audit:success", como em:

```
full_audit:success = open, opendir, write, unlink, rename, mkdir, rmdir, chmod, chown
```

(as opções foram uma única linha)

As opções que incluí no exemplo são open (ler um arquivo), opendir (ver os arquivos dentro de uma pasta), write (alterar um arquivo), unlink (deletar um arquivo), rename (renomear um arquivo), mkdir (criar um diretório), rmdir (remover um diretório), chmod (alterar as permissões de acesso de um arquivo) e chown (mudar o dono de um arquivo).

Você pode remover algumas destas opções deixando apenas as opções desejadas, ou ver uma lista completa das opções que podem ser incluídas no manual do `vfs_full_audit`, disponível no:

http://samba.org/samba/docs/man/manpages-3/vfs_full_audit.8.html

Continuando, especificamos as informações que desejamos que sejam incluídas no log, usando a opção `full_audit:prefix`. Aqui podemos utilizar as variáveis que mostrei no tópico sobre o compartilhamento [homes], como a `"%u"` (o nome do usuário), `"%I"` (o IP da máquina) e `"%S"` (o nome do compartilhamento onde foi feito o acesso ou a alteração). Não é necessário incluir a variável referente ao nome da máquina, pois o nome é incluído automaticamente:

```
full_audit:prefix = %u|%I|%S
```

Por padrão, o módulo `loga` não apenas os acessos e modificações, mas também um grande volume de mensagens de alerta e erros gerados durante a operação. A opção `full_audit:failure = none` evita que estas mensagens sejam logadas, fazendo com que o log fique muito mais limpo e seja mais fácil encontrar as opções que realmente interessam:

```
full_audit:failure = none
```

Concluindo, especificamos o nível dos alertas, entre os suportados pelo `syslog`, como em:

```
full_audit:facility = local5  
full_audit:priority = notice
```

Juntando tudo, temos:

```
vfs objects = full_audit  
full_audit:success = open, opendir, write, unlink, rename, mkdir, rmdir, chmod, chown  
full_audit:prefix = %u|%I|%S  
full_audit:failure = none  
full_audit:facility = local55  
full_audit:priority = notice
```

Esta configuração pode ser tanto incluída dentro da seção `[global]` (de forma que o log inclua os acessos e as alterações feitas em todos os compartilhamentos) quanto ser incluída apenas na configuração de um compartilhamento específico.

Com isso, o Samba vai passar a gerar os eventos referentes aos acessos. Falta agora configurar o `sysklogd`, para logar os eventos, gerando o arquivo de log que poderá ser consultado. Para isso, abra o arquivo `"/etc/syslog.conf"` e adicione a linha abaixo:

```
local5.notice /var/log/samba-full_audit.log
```

Note que o `"local5.notice"` corresponde aos valores informados nas opções `"full_audit:facility"` e `"full_audit:priority"`, enquanto o `"/var/log/samba-full_audit.log"` é o arquivo de log que será gerado.

Depois de concluída a configuração, reinicie os serviços e o log passará a ser gerado imediatamente:

```
# /etc/init.d/samba restart
# /etc/init.d/syslogd restart
```

Dentro do arquivo, você verá entradas contando a data e hora, o nome da máquina, o usuário, o IP da máquina, o nome do compartilhamento, a operação realizada e o nome do arquivo ou pasta onde ela foi realizada, como em:

```
Nov 18 15:21:15 m5 smb_d_audit: joao|192.168.1.23|arquivos|opendir|ok|.
Nov 18 15:21:29 m5 smb_d_audit: joao|192.168.1.23|arquivos|open|ok|r|addr.txt
Nov 18 15:21:34 m5 smb_d_audit: joao|192.168.1.23|arquivos|mkdir|ok|trabalho
Nov 18 15:21:36 m5 smb_d_audit: joao|192.168.1.23|arquivos|opendir|ok|trabalho
Nov 18 15:21:43 m5 smb_d_audit:
joao|192.168.1.23|arquivos|open|ok|w|trabalho/Samba.sxw
Nov 18 15:21:44 m5 smb_d_audit: joao|192.168.1.23|arquivos|open|ok|w|trabalho/foto.jpg
```

O log conterá entradas referentes a todos os usuários e máquinas, mas é fácil ver apenas as entradas referentes a um determinado usuário, compartilhamento, endereço IP ou outro parâmetro qualquer ao listar o arquivo pelo terminal usando o `grep`, que permite mostrar apenas as linhas contendo determinados trechos de texto, como em:

```
# cat /var/log/cat samba-full_audit.log | grep "joao|192.168.1.23"
```

(mostra os acessos provenientes do usuário joao e IP 192.168.1.23)

```
# cat /var/log/cat samba-full_audit.log | grep "|arquivos|"
```

(acessos feitos no compartilhamento "arquivos")

... e assim por diante. Você pode também direcionar a saída para um novo arquivo (ao invés de tentar lê-la pelo próprio terminal), como em:

```
# cat /var/log/cat samba-full_audit.log | grep "|arquivos|" > arquivos.log
```

Backends: smbpasswd ou tdbsam

As primeiras versões do Samba suportavam apenas o uso de senhas de texto puro, que eram transmitidas de forma não encriptada através da rede. Ainda é possível reverter a este sistema primitivo nas versões recentes do Samba usando a opção "encrypt passwords = no" no `smb.conf`, mas além de não trazer nenhuma vantagem, isso quebra a compatibilidade com todas as versões recentes do Windows, que não aceitam o envio de senhas em texto puro.

Para resolver o problema, foram criados diversos backends, que permitem armazenar senhas encriptadas e outras informações referentes aos usuários. Você pode escolher qual backend usar através da opção "passwd backend" do `smb.conf`. Vamos entender como eles funcionam.

O **smbpasswd** é o backend mais simples. Nele, as senhas são salvas no arquivo `"/etc/samba/smbpasswd"` e são transmitidas de forma encriptada através da rede, suportando o sistema NTLM, usado pelas versões contemporâneas do Windows.

A vantagem do `smbpasswd` é que ele é um sistema bastante simples. Embora encriptadas, as senhas são armazenadas em um arquivo de texto, com uma conta por linha.

Se você quer apenas configurar um servidor Samba para compartilhar arquivos e impressoras com a rede local, sem usá-lo como PDC, então o `smbpasswd` funciona bem. Ele é usado por padrão no Samba 3, de forma que se o arquivo `smb.conf` do seu servidor não contém a linha `"passdb backend ="` (como nos exemplos que vimos até aqui), você está usando justamente o `smbpasswd`.

Em seguida temos o **tdbsam**, que usa uma base de dados muito mais robusta, armazenada no arquivo `"/var/lib/samba/passdb.tdb"` (é justamente este arquivo que o script executado durante a instalação do pacote `samba` no Debian pergunta se deve ser criado).

O `tdbsam` oferece duas vantagens sobre o `smbpasswd`: oferece um melhor desempenho em servidores com um grande número de usuários cadastrados e oferece suporte ao armazenamento dos controles SAM estendidos usados pelas versões Server do Windows. O uso do `tdbsam` é fortemente recomendável caso seu servidor tenha mais do que algumas dezenas de usuários cadastrados ou caso você pretenda usar seu servidor Samba como PDC da rede (veja mais detalhes a seguir). Ele é também um pré-requisito caso você precise migrar um domínio NT já existente para o servidor Samba.

Ao usar uma versão recente do Samba, ativar o uso do `tdbsam` é bastante simples, basta incluir a linha `"passdb backend = tdbsam"` na seção `[global]` do `smb.conf`, como em:

```
[global]
netbios name = Sparta
workgroup = Grupo
server string = Servidor
encrypt passwords = true
wins support = yes
preferred master = yes
os level = 100
enable privileges = yes
passdb backend = tdbsam
```

Embora o arquivo de senhas seja diferente, o comando para cadastrar os usuários no Samba continua sendo o mesmo:

```
# smbpasswd -a usuario
```

Isso acontece por que, ao ser executado, o `smbpasswd` verifica a configuração presente no `smb.conf` e assim realiza as operações necessárias para cadastrar os usuários no backend utilizado.

A principal dica é que, ao utilizar o `tdbsam`, você deve adicionar a linha `"passdb backend = tdbsam"` no `smb.conf` logo no início da configuração, antes de começar a cadastrar os usuários no servidor, caso contrário o `smbpasswd` cadastrará os usuários no `smbpasswd` e

you will need to re-register them again to update the tdbSAM database later. In many cases, a script included in the distribution can take care of the conversion automatically, but it is better not to rely on it.

To verify if users are registered in the tdbSAM database, use the command "**pdbedit -Lw**" (as root). It should return a list containing all registered users.

Next we have **mysqSAM** and **ldapsAM**, where accounts and passwords are stored in, respectively, a MySQL server and an LDAP server. The use of MySQL together with Samba is not very common, but LDAP is growing in large networks.

A big advantage is that the database can be accessed by several servers, without the need to replicate the password file manually (using rsync, for example). This is very useful in the case of very large networks where user authentication is divided among several servers. In this configuration, the PDC shares the workload with a set of BDCs (backup domain controllers), which can be either Samba or Windows servers. BDCs are subordinate to the PDC, but all have access to the same database with users stored in the LDAP server, which avoids synchronization problems between them.

In general, a single PDC using tdbSAM as backend serves up to 250 clients. This limit is not related to the use of tdbSAM, but to practical issues related to network performance. It can be higher or lower in practice, depending on network speed (100 or 1000 megabits), server hardware and network load. From here, it makes sense to migrate to an LDAP database and add secondary BDC servers.

Portas e firewall

Samba is a server designed to be used within the local network, or the intranet. It is difficult to imagine a situation where you would want to make a Samba server available on the internet. It is relatively common to find machines or even Windows servers where file sharing and printers are available to the world due to some oversight by the owner, but it is almost impossible to find someone who does it intentionally.

There are many ways to prevent the Samba server from being available on the internet. In a server with two network cards, the simplest is to configure the firewall to block all connections from the network card connected to the internet and/or add the line "**interfaces = eth1**" (where "eth1" is the local network card) in the [global] section of smb.conf, which makes Samba listen only on the specified interface.

Another option that can be used is "hosts allow", which allows you to specify a range of addresses from which the server will accept requests. Limiting access to the local network, you guarantee that it will not be accessed from the internet. In this case, you would add the line "**hosts allow = 192.168.0.**" (where 192.168.0. is the local network range) in the [global] section. You can even combine the three things (the firewall and the two restrictive rules) if you want, because security is never too much.

In the case of the local network, the firewall is not always necessary, because in small networks, normally you know the users. In larger networks, however, the scenario is more "each for itself" and a more careful configuration is necessary. The ideal is to activate the firewall and keep only the services intentionally available, minimizing the chance of someone getting access to the server through some service that you didn't know was active.

As portas usadas pelo Samba, que precisam ficar abertas na configuração do Firewall são:

137/udp: Usada pelo Daemon nmbd, responsável pela navegação nos compartilhamentos de rede.

138/udp: Também usada pelo nmbd, desta vez para a resolução dos nomes das máquinas da rede

139/tcp: Usada pelo daemon smb, o componente principal do Samba, responsável pelo compartilhamento de arquivos e impressoras.

445/tcp: Esta porta é usada pelos clientes Windows 2000, XP e Vista para navegação na rede. Eles utilizam o protocolo CIFS, no lugar do antigo protocolo NetBIOS.

Um exemplo de regras do Iptables que você poderia incluir no seu script de firewall para mantê-las abertas é:

```
iptables -A INPUT -p udp --dport 137 -j ACCEPT
iptables -A INPUT -p udp --dport 138 -j ACCEPT
iptables -A INPUT -p tcp --dport 139 -j ACCEPT
iptables -A INPUT -p tcp --dport 445 -j ACCEPT
```

Samba, parte 3: Usando o Samba como PDC

Em uma pequena rede, manter as senhas dos usuários sincronizadas entre as estações Windows e o servidor Samba não chega a ser um grande problema. No entanto, em redes de maior porte, isso pode se tornar uma grande dor de cabeça e passar a consumir uma boa parte do seu tempo.

Para solucionar o problema, existe a opção de usar o servidor Samba como um controlador primário de domínio (PDC), onde ele passa a funcionar 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. Isso 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 em um controlador de domínio. A seção "global" deve conter as linhas "domain master = yes", "domain logons = yes" e "logon script = netlogon.bat" e (importante) **não** deve conter a linha "invalid users = root", pois precisaremos usar a conta de root no Samba ao configurar os clientes. É preciso ainda adicionar um compartilhamento chamado "netlogon", que conterá o script de logon que será executado pelas estações.

É necessário também que o modo de segurança esteja configurado em nível de usuário (security = user) e que o uso de senhas encriptadas esteja ativado (encrypt passwords = yes). Na verdade, não é obrigatório incluir estas duas linhas, pois estes valores são usados por default pelo Samba 3, mas é sempre interessante usá-los em exemplos e modelos de configuração para fins didáticos e para deixar claro que o arquivo não deve ter opções que conflitem com elas.

Embora não seja obrigatório, é fortemente recomendável ativar o uso do tdbsam, adicionando a linha "passdb backend = tdbsam". Usar o smbpasswd em um PDC oferece várias desvantagens. A principal delas é que o smbpasswd armazena um conjunto bastante incompleto de atributos referentes aos usuários, de forma que atributos como o SID (um código de identificação único a cada usuário, usado como verificação de segurança) ficam em branco ou são gerados dinamicamente durante os acessos, o que pode quebrar o suporte aos roaming profiles em algumas situações. O uso do tdbsam soluciona estes problemas.

Este é um exemplo de arquivo de configuração do Samba para um controlador de domínio. Ele não contém as configurações para compartilhamento de impressoras, lixeira e outras opções que você pode adicionar (juntamente com os compartilhamentos desejados) depois de testar a configuração básica:

```
[global]
workgroup = Dominio
netbios name = GDH
server string = Samba PDC

domain master = yes
domain logons = yes
logon script = netlogon.bat

security = user
encrypt passwords = yes
enable privileges = yes
passdb backend = tdbsam

preferred master = yes
local master = yes
os level = 100
wins support = yes

[netlogon]
comment = Servico de Logon
path = /var/samba/netlogon
read only = yes
browseable = no

[homes]
valid users = %S
create mask = 0700
directory mask = 0700
browseable = no
```

Acostume-se a sempre rodar o comando **"testparm"** depois de fazer alterações no arquivo, 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", como em:

```
$ testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[netlogon]"
Processing section "[homes]"
Loaded services file OK.
Server role: ROLE_DOMAIN_PDC
```

As linhas "preferred master = yes", "local master = yes" e "os level = 100" fazem com que o servidor assuma também a função de master browser da rede. É comum que o PDC acumule também a função de master browser, mas na verdade uma coisa não tem relação com a outra. Você pode remover as três linhas e configurar outra máquina para assumir a função de master browser se preferir.

Depois de configurar o arquivo, verifique se a conta root do sistema foi cadastrada no Samba e se as senhas estão iguais. Caso necessário, use o comando **"smbpasswd -a root"** para cadastrar a conta de root no Samba. Aproveite para criar a pasta `"/var/samba/netlogon"` e configurar 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 um usuário de teste "joao", use os comandos:

```
# adduser joao
# smbpasswd -a joao
```

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

```
# mkdir /home/joao/profile.pds
```

Ao rodar este comando como root, não se esqueça de ajustar as permissões da pasta, de forma que o usuário seja o dono:

```
# chown -R joao.joao /home/joao/profile.pds
```

Aproveite e crie a pasta "profile.pds" dentro do diretório /etc/skel, de forma que ela seja criada automaticamente dentro do home dos usuários que criar daqui em diante:

```
# mkdir /etc/skel/profile.pds
```

Além das contas para cada usuário, é preciso cadastrar também uma conta (bloqueada, e por isso 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 "alesia" por exemplo, é preciso criar um login de máquina com o mesmo nome:

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

Note que nos dois primeiros comandos é adicionado um "\$" depois do nome, que indica que estamos criando uma conta 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). Estas contas de máquina são chamadas de "trusted accounts" ou "trustee".

Lembre-se de que para usar este comando o arquivo "/etc/shells" deve conter a linha "/bin/false". Em caso de erro ao adicionar a máquina, use o comando abaixo para adicionar a linha e tente novamente (este comando só funciona se executado diretamente usando o root, não funciona se executado usando o sudo):

```
# echo "/bin/false" >> /etc/shells
```

Se preferir, você pode adicionar as contas de máquina dentro de um grupo do sistema ("maquinas" ou "machines" por exemplo). Neste caso, crie o grupo usando o comando "groupadd" e use o comando abaixo para criar as contas de máquina já incluindo-as no grupo:

```
# useradd -g maquinas -d /dev/null -s /bin/false alesia$
```

Por último, é necessário criar o arquivo "/var/samba/netlogon/netlogon.bat", um script que é lido e executado pelos clientes ao fazer logon. Você pode fazer muitas coisas através dele, mas um exemplo de arquivo funcional é:

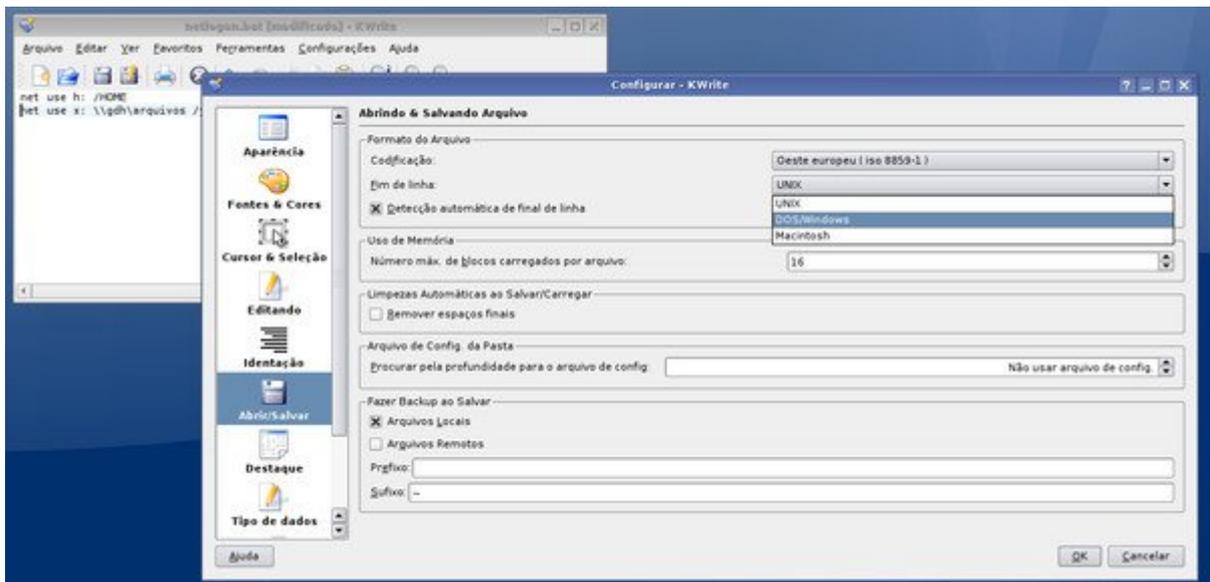
```
net use h: /HOME  
net use x: \\gdh\arquivos /yes
```

Este script faz com que a pasta home de cada usuário (compartilhada pelo Samba através da seção "homes") 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: \\gdh\arquivos /yes" faz com que o compartilhamento "arquivos" (que precisaria ser configurado no smb.conf) seja mapeado como o drive "X:" nos clientes. Lembre-se que o "gdh" dentro do netlogon.bat deve ser

substituído pelo nome do seu servidor Samba, configurado na opção "netbios name =" do smb.conf.

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 (que é o padrão da maioria dos editores de texto do Linux). 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: "Configurações > 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**, ou perfis móveis), fornecendo-os à estação no momento em que o usuário faz logon. Isso 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, ativar os perfis móveis 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. Isso 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 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. Quando, por exemplo, o "joao" faz logon é montado o compartilhamento "\\gdh\profiles\joao". 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
```

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

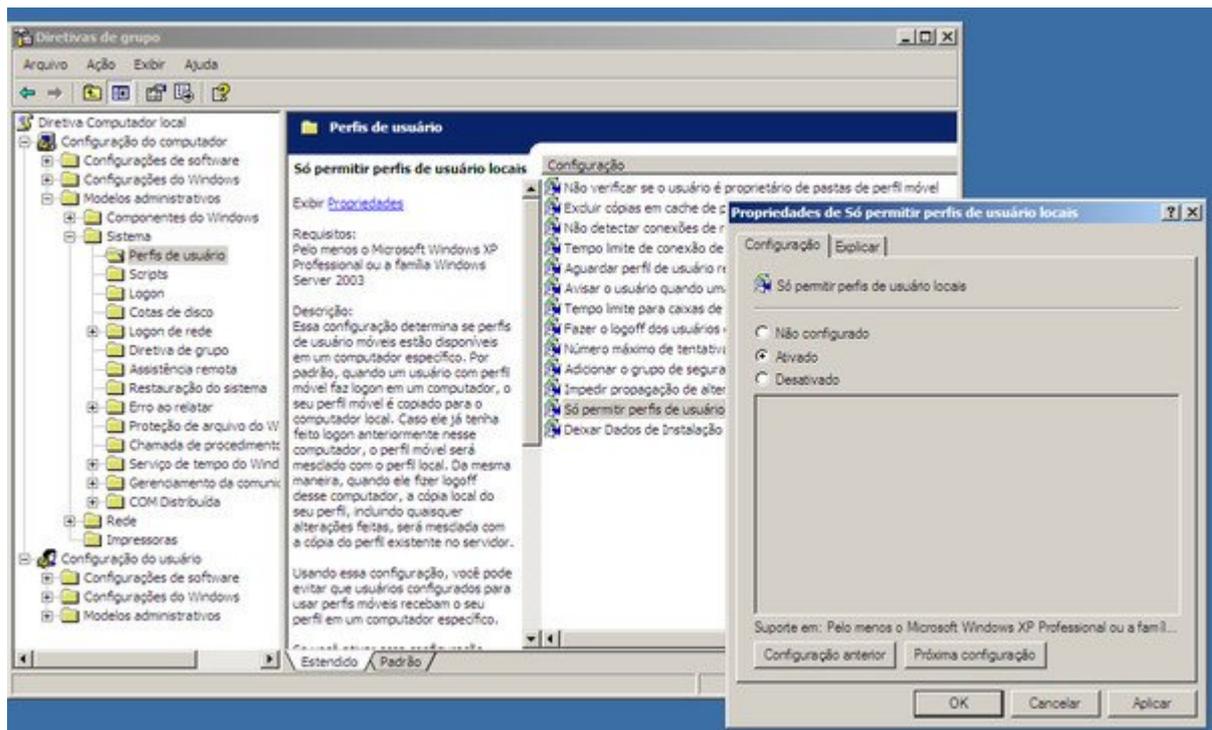
```
# mkdir /var/profiles
# chmod 1777 /var/profiles
```

Cada usuário passa a ter uma pasta pessoal dentro da pasta (`"/var/profiles/joao"`, por exemplo) 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 perfil.

As estações Windows 2000 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".

No Windows XP o default foi alterado e o sistema tenta usar o perfil móvel por padrão, exibindo uma mensagem de erro (repetida a cada logon) caso o recurso não esteja disponível no servidor. Para eliminar as mensagens de erro é necessário desativar o uso dos perfis móveis, o que é feito através do utilitário "**gpedit.msc**", que pode ser chamado através do Iniciar > Executar (é necessário estar logado localmente, usando uma conta com privilégios administrativos).

Dentro dele, acesse a opção "Configuração do computador > Modelos administrativos > Sistema > Perfis de usuário > Só permitir perfis de usuário locais" e mude a opção de "Não configurado" para "Ativado" (esta alteração precisa ser repetida em todas as máquinas):



Aqui vai mais um exemplo de configuração, incluindo a configuração para uso como PDC, o compartilhamento netlogon, suporte a perfis móveis e compartilhamento de impressoras:

[global]

```
netbios name = Byzantium
workgroup = Dominio
server string = Servidor PDC
domain master = yes
domain logons = yes
logon script = netlogon.bat
logon home = \\%L\%U\profiles
logon path = \\%L\profiles\%U
```

```
security = user
encrypt passwords = yes
enable privileges = yes
passdb backend = tdbsam
preferred master = yes
local master = yes
os level = 100
wins support = yes
```

```
printing = cups
load printers = yes
enable privileges = yes
```

[printers]

```
path = /var/spool/samba
print ok = yes
```

```
guest ok = yes  
browseable = yes
```

```
[print$]  
path = /var/smb/printers  
read only = yes  
write list = gdh  
inherit permissions = yes
```

```
[netlogon]  
comment = Servico de Logon  
path = /var/samba/netlogon  
read only = yes  
browseable = no
```

```
[profiles]  
path = /var/profiles  
writeable = yes  
browseable = no  
create mask = 0600  
directory mask = 0700
```

```
[homes]  
valid users = %S  
create mask = 0700  
directory mask = 0700  
browseable = no
```

```
[arquivos]  
path = /mnt/hda2  
writable = no  
write list = +arquivos
```

Com o servidor Samba configurado, falta o mais importante, que é configurar os clientes para fazerem logon no domínio. Ao usar um PDC, surge a necessidade de cadastrar as máquinas no domínio, para só então os usuários cadastrados poderem utilizar as máquinas. É possível cadastrar tanto máquinas Windows quanto máquinas Linux no domínio, vamos agora às peculiaridades de cada sistema.

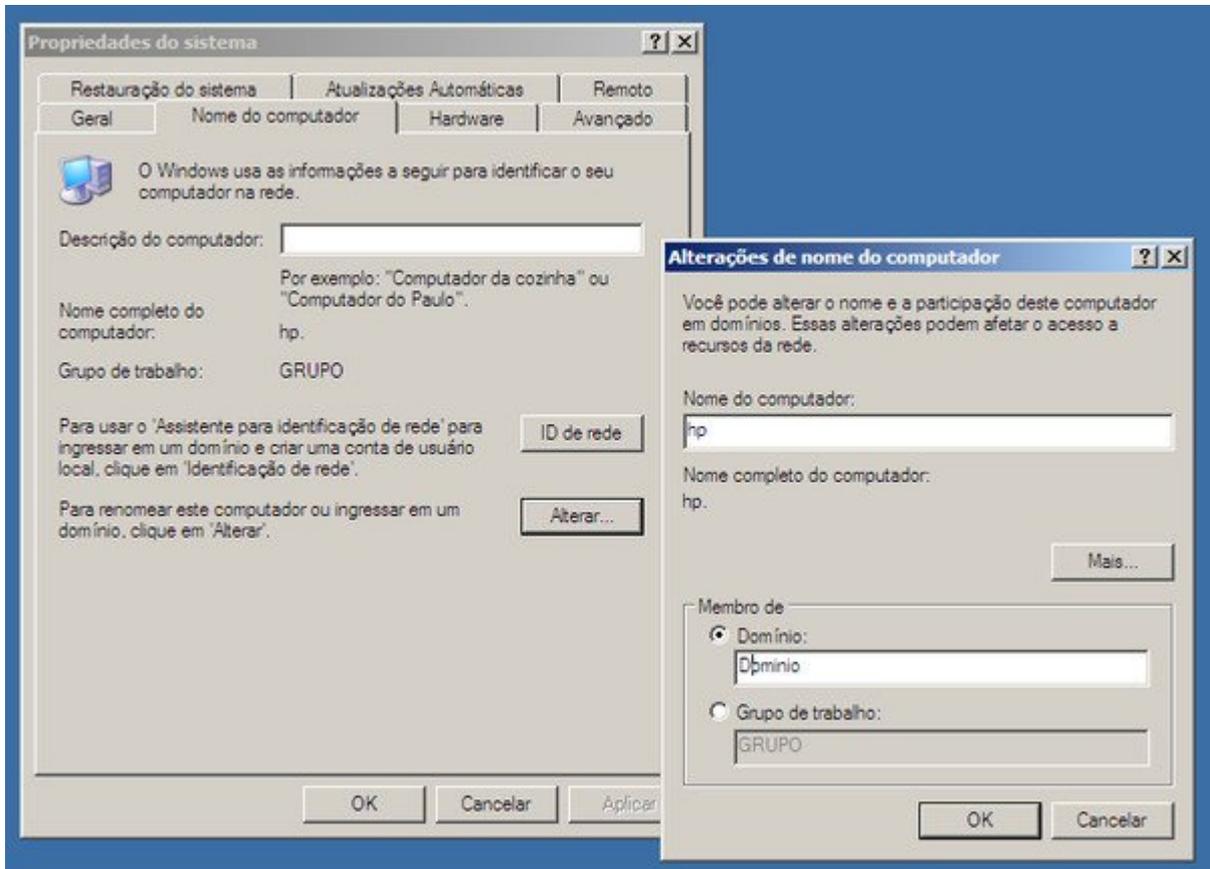
Logando Clientes Windows

Nem todas as versões do Windows suportam o uso de um domínio. 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 (Miserable Edition), assim como no Vista Starter, Vista Home Basic e Vista Home Premium, de forma a pressionar as empresas a comprarem as versões mais caras.

A configuração muda de acordo com a versão do Windows:

No **Windows XP Professional**, acesse o "Painel de Controle > Sistema > Nome do Computador" e use a opção "Alterar...". No menu seguinte, defina o nome da máquina

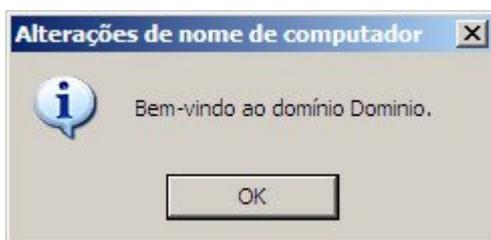
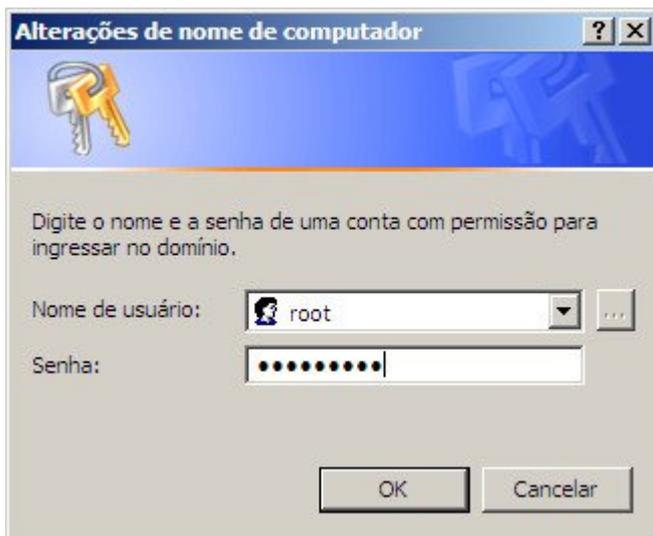
(que precisa ser um dos logins de máquinas adicionados na configuração do Samba) e o nome do domínio, que é definido na opção "workgroup =" do smb.conf. Para ter acesso a esta opção você deve estar logado como administrador:



Nunca é demais lembrar que o "Nome do computador" fornecido na opção deve corresponder a uma das contas de máquinas cadastradas no servidor Samba, usando os três comandos que citei anteriormente. Para cadastrar a máquina "hp", por exemplo, você usaria (no servidor, como root) os comandos abaixo:

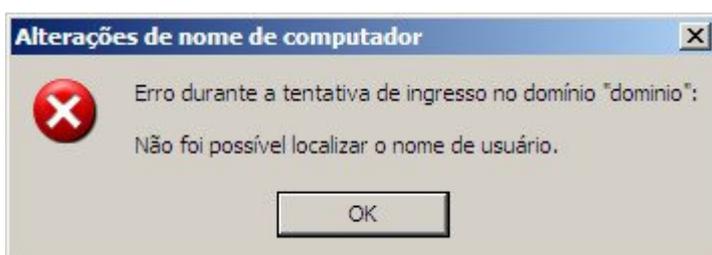
```
useradd -d /dev/null -s /bin/false hp$  
passwd -l hp$  
smbpasswd -a -m hp
```

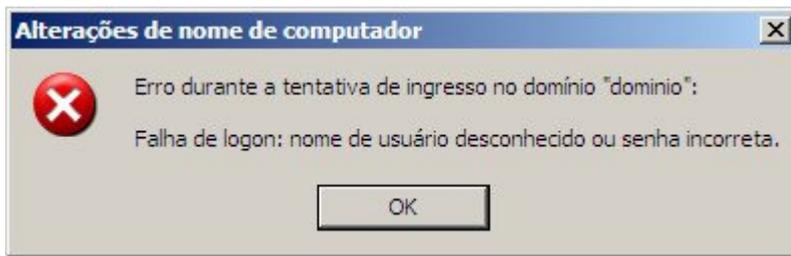
Na tela de identificação que será aberta a seguir, logue-se como "root", com a senha definida no servidor Samba. É normal que a conexão inicial demore um ou dois minutos. Se tudo der certo, você é saudado com a mensagem "Bem-vindo ao domínio Dominio" (onde o "Domínio" é o nome definido na opção "workgroup" do smb.conf):



Fornecer a senha de root do servidor ao cadastrar o cliente no domínio, prova que quem está fazendo a operação é o administrador, ou alguém autorizado por ele. Se qualquer um pudesse adicionar e remover máquinas do domínio, ele não seria muito diferente de um grupo de trabalho e a configuração perderia todo o sentido. Se você não gostou da idéia de usar a senha de root para cadastrar as máquinas, é possível também outorgar o privilégio a uma outra conta através do comando "net", como veremos a seguir.

Aqui temos duas mensagens de erros comuns ao tentar cadastrar a máquina no domínio. A primeira (Não foi possível localizar o nome de usuário) aparece quando o nome da máquina não foi cadastrado no servidor Samba como uma conta de máquina e a segunda (Falha de logon) indica que a conta de root não foi cadastrada no Samba (smbpasswd -a root), que a senha informada no cliente está incorreta ou que o Samba não está sendo capaz de utilizar a conta de root devido à presença da linha "invalid users = root" no smb.conf:





Quando a máquina passa a fazer parte do domínio, é criada uma "relação de confiança" entre ela e o servidor. Uma senha (chamada de "machine trust account password") é usada pela máquina para comprovar sua identidade ao contatar o servidor de domínio. Esta é uma senha interna, gerada automaticamente pelo sistema durante a conexão inicial.

Depois de reiniciar a estação, aparece a opção "Efetuar logon em: DOMINIO" 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, mas neste caso perde-se o acesso aos recursos relacionados ao domínio e é usado o perfil do usuário local:



Para remover a máquina do domínio, é preciso acessar a mesma opção e mudar a opção de "Membro de Domínio:" para "Membro de Grupo de trabalho:". O sistema solicita novamente a senha do servidor, como uma forma de comprovar que o usuário está autorizado a realizar a operação. Isso evita que os usuários da rede desfaçam a configuração, removendo as máquinas do domínio.

Para confirmar se os clientes estão realmente efetuando logon no servidor, use o comando "**smbstatus**" (no servidor). Ele retorna uma lista dos usuários e máquina logadas, como em:

Samba version 3.0.14a-Debian

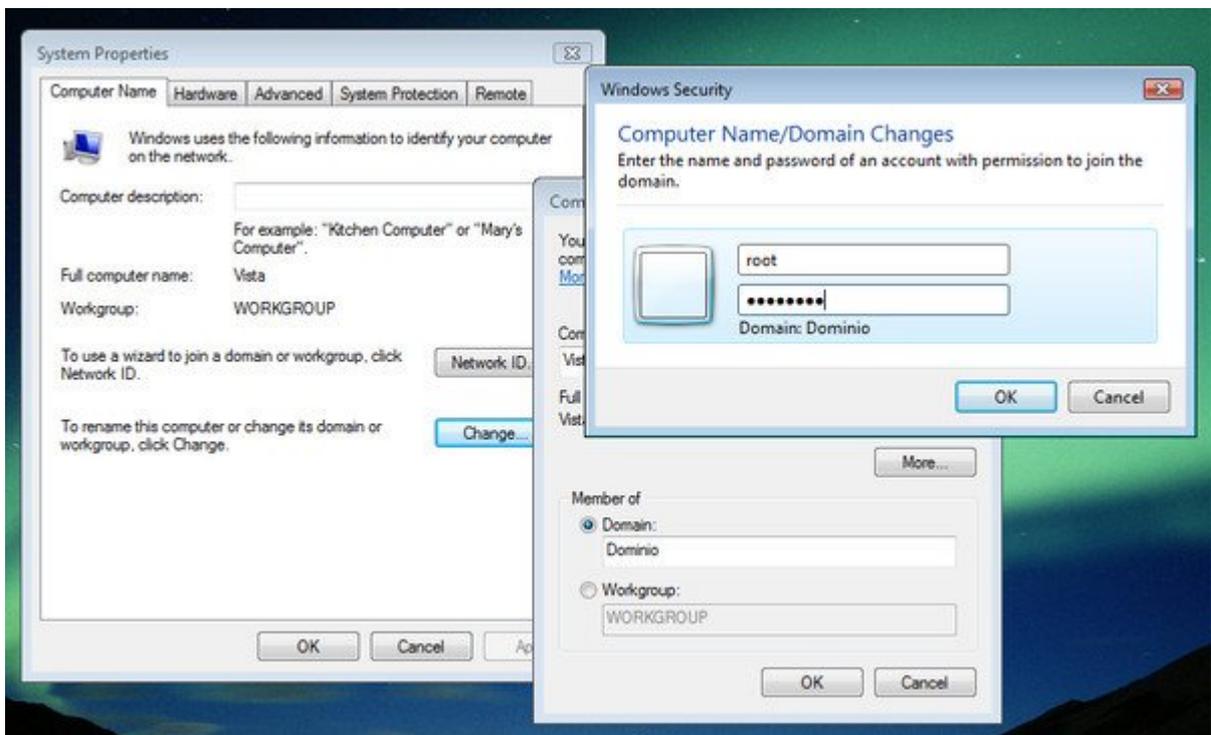
PID Username Group Machine

4363 joao joao athenas (192.168.0.34)

Service pid machine Connected at

joao 4363 athenas Sat Jul 9 10:37:09 2005

No Windows Vista, a opção de adicionar a máquina ao domínio está no Painel de Controle > Sistema > Configurações avançadas do sistema (na lista à esquerda) > Nome do Computador > Alterar:



A forma como você escolhe se quer se logar ao domínio ou fazer um login na máquina local na tela de login do Vista segue uma lógica um pouco curiosa.

Depois que a máquina é adicionada ao domínio, a tela de login mostra a opção de fazer logon no domínio, onde o último login utilizado fica pré-selecionado. Para usar outro login, é necessário clicar no botão "Trocar Usuário" e fornecê-lo na tela seguinte. Entretanto, não existe uma opção para fazer logon na máquina local. Para isso, é necessário especificar o nome da máquina seguido pelo nome do usuário no campo de login, como em: Vista\gdh. Outra opção é usar um "." antes do nome do usuário, como em ".\gdh".

No Windows 2000, o procedimento é basicamente o mesmo do Windows XP, muda apenas a localização da opção, que está disponível no "Meu Computador > Propriedades > Identificação de rede > Propriedades".

No Windows 98 ou ME: Ao contrário do XP Home, XP Starter, Vista Starter e Vista Home, as máquinas com o Windows 98 ou ME podem ser adicionadas ao domínio. Entretanto, elas participam dentro de um modo de compatibilidade, onde podem acessar os compartilhamentos, mas não têm acesso ao recurso de perfis móveis, por exemplo. Para cadastrar a máquina, comece logando-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 do Windows (para a instalação dos componentes necessários) e reiniciar a máquina.

Cadastrando as máquinas sem usar a conta de root

Normalmente, você deve fornecer a senha de root ao inserir cada máquina no domínio. Fornecer a senha de root é justamente uma prova de que você é realmente o administrador do servidor e está autorizado a cadastrar as máquinas. Esta é a forma mais simples de trabalhar, mas muitos torcem o nariz para a idéia, temendo abrir uma brecha para ataques.

É possível evitar a necessidade de usar a conta de root ao cadastrar as máquinas criando uma conta especial, com privilégios para adicionar máquinas ao domínio, de forma similar ao que fizemos ao configurar o fornecimento automático de drivers de impressão.

Para isso, usamos novamente o comando "net", adicionando agora o privilégio "SeMachineAccountPrivilege" ao usuário que terá permissão para adicionar as máquinas no domínio. Se o servidor se chama "athenas" e o usuário se chama "gdh", o comando seria:

```
# net -S localhost -U root -W ATHENAS rpc rights grant 'ATHENAS\gdh'  
SeMachineAccountPrivilege
```

(todo o comando forma uma única linha)

Este comando deve ser executado em um prompt do próprio servidor (você pode se logar via SSH, por exemplo). Ele vai solicitar a senha de root e exibir uma mensagem de confirmação. A partir daí você pode usar o login "gdh" e a senha cadastrada ao cadastrar as máquinas no lugar da conta root.

Lembre-se de que, para adicionar os privilégios, você deve comentar ou remover a linha "invalid users = root" e adicionar a linha "enable privileges = yes" na seção [global] do smb.conf, como vimos no tutorial sobre compartilhamento de impressoras no Samba.

Ajustando as permissões locais

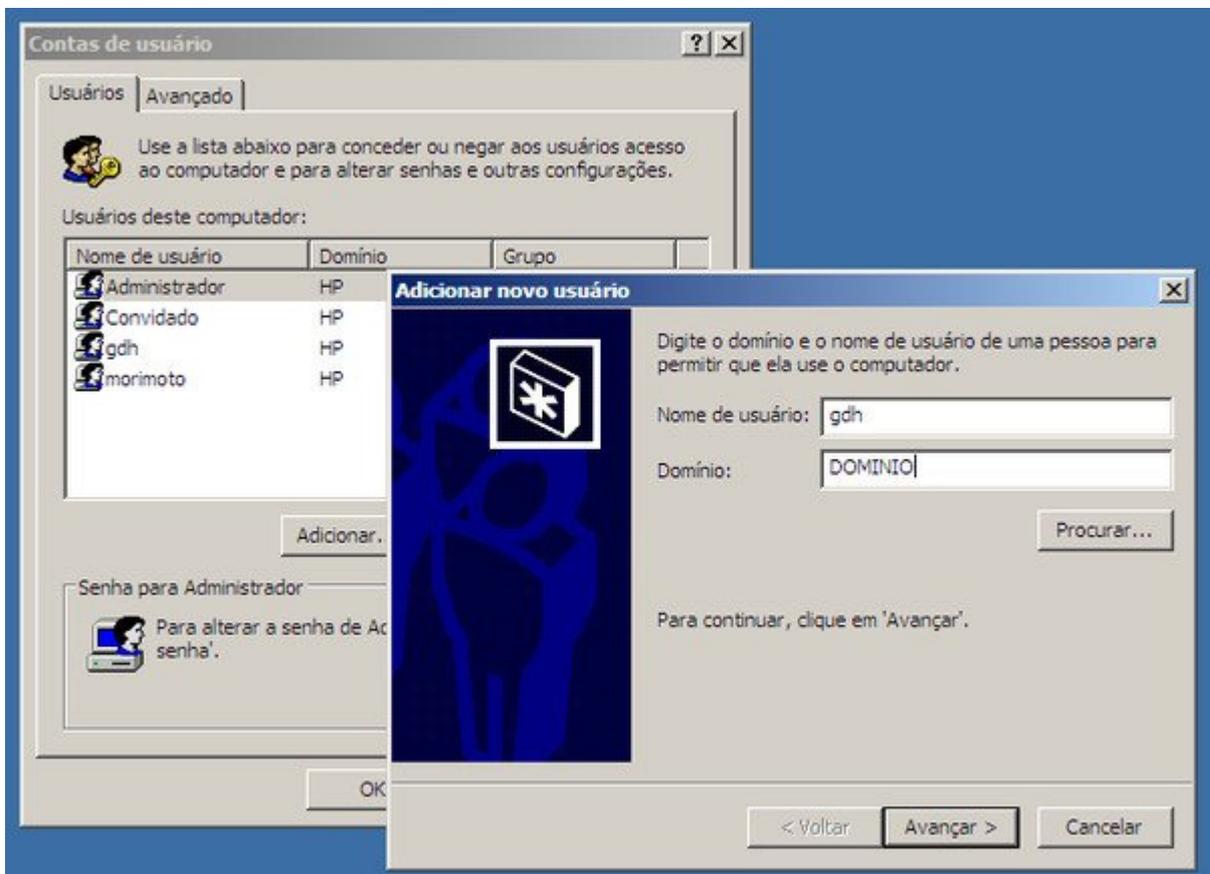
Ao adicionar a máquina Windows ao domínio, é criada uma distinção entre as contas locais e as contas de domínio. Quando o usuário se loga na estação Windows usando uma das contas cadastradas no servidor, ele é na verdade logado (na estação local) usando uma conta limitada, onde ele não tem permissão para compartilhar arquivos, para alterar as configurações da rede, nem para alterar a maior parte das configurações do sistema.

Em muitas situações, é exatamente isso que você quer, mas em outras isso pode ser um grande problema, já que o usuário não conseguirá compartilhar pastas com outros usuários da rede, por exemplo. Veja que a aba de compartilhamento sequer fica disponível nas propriedades da pasta:

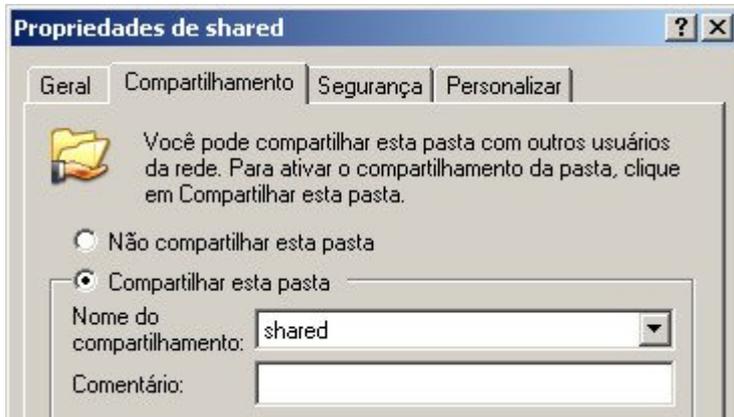


Para mudar isso, é necessário ajustar as permissões da máquina local, de forma que a conta do domínio tenha permissão para alterar as configurações. Para isso, logue-se localmente na estação Windows, usando uma conta com privilégios administrativos e acesse o "Painel de controle > Contas de usuário".

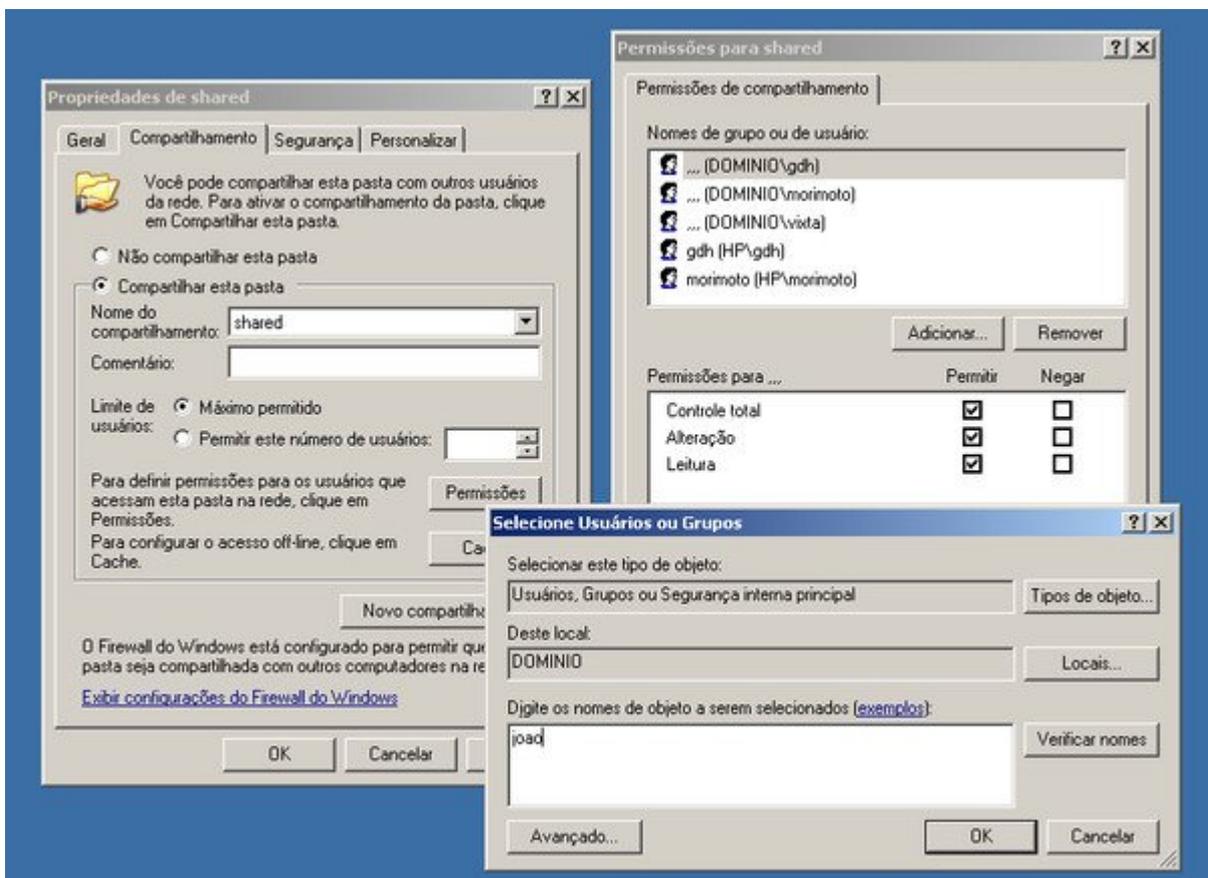
Clique no "Adicionar" e especifique o login do usuário e o nome do domínio e na tela seguinte especifique o nível de permissão na máquina local (Administrador, Usuário avançado, etc.). Você pode adicionar outros usuários se desejar:



Faça logoff e logue-se novamente no domínio com a conta que foi cadastrada. Se você a cadastrou com privilégios administrativos, você notará que a aba de compartilhamento voltou a aparecer e o acesso às demais configurações foi destravado. Com isso o usuário assume o controle de sua máquina local e pode criar compartilhamentos e alterar as demais configurações:



Inicialmente, os compartilhamentos aparecerão no ambiente de rede, mas usuários de outras máquinas (também cadastradas no domínio) não conseguirão acessá-los, recebendo uma mensagem de permissão negada. Para solucionar este último problema, acesse as permissões da pasta e adicione os usuários do domínio que terão permissão para acessá-la e as permissões de cada um:



Note que esta configuração é necessária apenas se você quiser que os usuários das estações possam criar compartilhamentos locais. Outra opção é simplesmente adicionar

compartilhamentos no servidor e orientar os usuários a usarem os compartilhamentos criados para compartilharem os arquivos desejados. Centralizar todos os compartilhamentos no servidor Samba é mais seguro e facilita bastante os backups, já que você precisará apenas em fazer backup dos arquivos do servidor.

Continuando, é possível também criar usuários administrativos, com permissão para alterar o dono e as permissões dos arquivos colocados nos compartilhamentos do próprio servidor. Isso é feito usando o comando "net", o mesmo que utilizamos para permitir que o usuário possa dar upload dos drivers de impressão e possa adicionar máquinas ao domínio.

Os três privilégios relacionados são

SeDiskOperatorPrivilege: Permite que o usuário altere as permissões de acesso dos compartilhamentos e arquivos dentro deles.

SeRestorePrivilege: Permite que o usuário altere o dono dos arquivos e pastas, transferindo a posse para outro usuário (exceto ele mesmo)

SeTakeOwnershipPrivilege: Permite que o usuário assuma para si a posse de arquivos e pastas, complementando o SeRestorePrivilege.

Se o servidor se chama "athenas" e o usuário que receberá os privilégios se chama "gdh", os comandos para fornecer os três privilégios (a serem executados em um terminal do servidor) seriam:

```
# net -S localhost -U root -W ATHENAS rpc rights grant 'ATHENAS\gdh'  
SeDiskOperatorPrivilege  
# net -S localhost -U root -W ATHENAS rpc rights grant 'ATHENAS\gdh'  
SeRestorePrivilege  
# net -S localhost -U root -W ATHENAS rpc rights grant 'ATHENAS\gdh'  
SeTakeOwnershipPrivilege
```

Não é preciso dizer que em uma grande rede estes privilégios devem ser atribuídos apenas a outros administradores ou a usuários de sua inteira confiança, já que eles permitem acesso quase que irrestrito aos arquivos no servidor.

Para listar os privilégios atribuídos a cada usuário, use o comando:

```
# net -S localhost -U% rpc rights list accounts
```

Para remover um determinado privilégio, é usado o mesmo comando que para adicionar, apenas substituindo o "grant" por "revoke", como em:

```
# net -S localhost -U root -W ATHENAS rpc rights revoke 'ATHENAS\gdh'  
SeTakeOwnershipPrivilege
```

Logando Clientes Linux

Naturalmente, também é possível logar clientes Linux no domínio, já que o Samba pode ser usado como cliente tanto de um PDC Samba quanto de um PDC Windows. Isso permite que a estação Linux acesse os recursos do domínio normalmente e utilize o PDC como um servidor de autenticação na hora de compartilhar arquivos com a rede, da mesma forma que as máquinas Windows. Com isso, você elimina a necessidade de cadastrar os logins de usuários em todas as máquinas que precisarem compartilhar arquivos, já que todo o processo de autenticação é centralizado no servidor.

O primeiro passo é cadastrar o nome da máquina no servidor PDC, usando os três comandos que já vimos: "useradd -d /dev/null -s /bin/false nome\$", "passwd -l nome\$" e "smbpasswd -a -m nome". No caso dos clientes Linux, vale o nome definido durante a instalação do sistema, que fica armazenado dentro do arquivo "/etc/hostname".

Esta é a única configuração que precisa ser feita no servidor. Os passos seguintes são feitos no próprio cliente.

Comece fazendo uma instalação normal do Samba, instalando os pacotes "samba" e "samba-client" (smbclient) através do gerenciador de pacotes, da mesma forma que faria ao instalar um servidor Samba para a rede.

É necessário cadastrar pelo menos uma conta de usuário no Samba (da estação), usando a mesma senha definida no sistema, usando o comando smbpasswd, como em:

```
# smbpasswd -a joao
```

Com o Samba instalado, edite o arquivo smb.conf, deixando-o como este modelo:

```
[global]
netbios name = M5
workgroup = Dominio
security = domain
encrypt passwords = yes

password server = 192.168.1.254
username map = /etc/samba/smbusermap

[arquivos]
path = /home/arquivos
writable = yes
```

A seção global deve conter as linhas "security = domain" (como citei anteriormente, este é o nível de segurança que permite que o Samba atue como cliente de um PDC), "encrypt passwords = yes" e a linha "password server =" que indica o endereço IP (ou nome netbios) do servidor PDC. É importante também que a linha "workgroup" inclua o nome correto do domínio e a linha "netbios name" contenha o nome da máquina, como cadastrado no servidor e salvo no arquivo "/etc/hostname".

Você pode incluir também os compartilhamentos de arquivos e impressoras desejados, como no caso do compartilhamento [arquivos] que incluí no exemplo.

Depois de salvar o arquivo e reiniciar o serviço, é hora de adicionar a máquina ao domínio, o que é feito usando o comando abaixo:

```
# net join -U root  
Password:  
Joined domain DOMINIO.
```

A senha de root solicitada é a senha de root cadastrada no servidor, que é checada ao cadastrar a estação como uma forma de provar que você é o administrador da rede. Você pode também criar um usuário administrativo com poderes para adicionar as máquinas ao domínio (evitando assim o uso da conta de root), dando a ela o privilégio "SeMachineAccountPrivilege", como vimos no tópico anterior.

Se o comando exibir a mensagem "Joined domain DOMINIO." sem solicitar a senha, rode-o novamente, pois isso acontece quando (por qualquer motivo) ele não conseguiu contactar o servidor. Se ele reclamar que a senha está incorreta, ou exibir um erro de permissão, verifique a configuração do servidor. Isso acontece, por exemplo, quando a linha "invalid users = root" está presente na configuração.

Uma vez inserida no domínio, a instância do Samba rodando na estação passará a encaminhar todos os pedidos de autenticação para o servidor. Se o servidor autoriza o acesso, então o servidor Samba local permite o acesso ao compartilhamento. Com isso, um novo usuário cadastrado no servidor PDC, ganha acesso também aos compartilhamentos do estação, sem que você precise cadastrá-lo duas vezes.

Para que isso funcione é necessário duas coisas. Em primeiro lugar, é necessário especificar o endereço IP ou nome do servidor, para que a estação consiga contactá-lo, o que é feito através da opção "password server", como já vimos.

O segundo passo é criar um arquivo com um mapa dos usuários na estação. O arquivo pode ser armazenado em qualquer pasta, mas você precisa especificar sua localização corretamente na opção "username map" do smb.conf, como em:

```
username map = /etc/samba/smbusermap
```

Este arquivo relaciona os logins cadastrados no servidor PDC com a conta cadastrada no no servidor Samba local, explicando a ele como acessar os arquivos no sistema uma vez que o acesso é autorizado pelo PDC. Sem isso, o sistema bloqueia o acesso, já que as contas cadastradas no PDC não existem localmente.

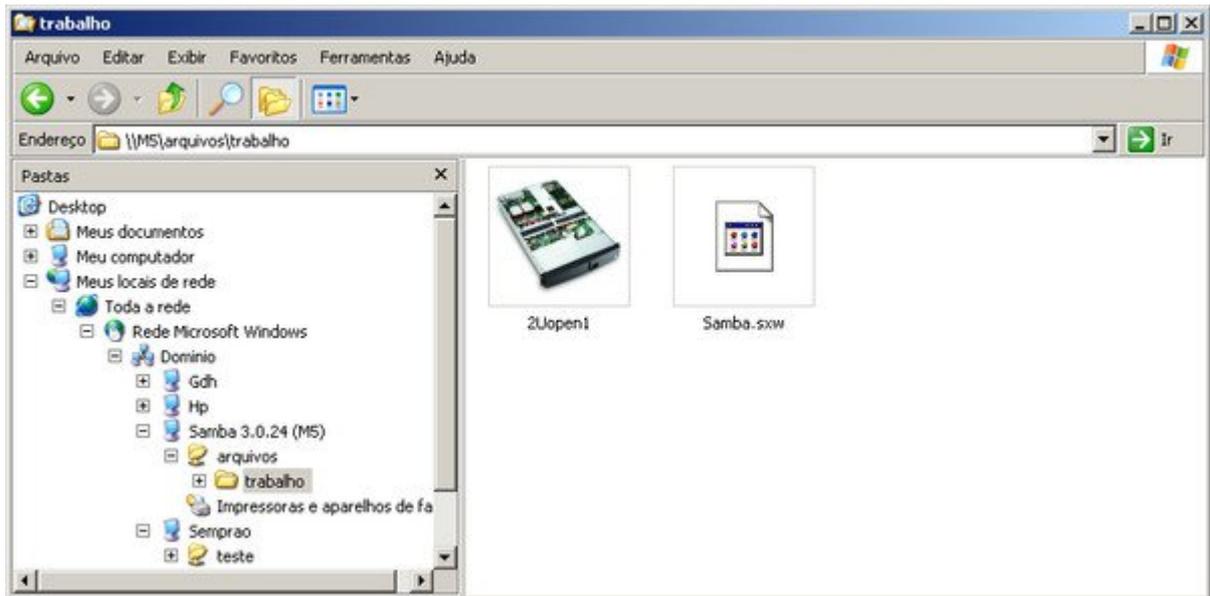
O arquivo com o username map segue uma estrutura muito simples, onde você especifica uma conta por linha, sempre seguindo a sintaxe "conta_local = nome_do_dominio\conta_no_dominio", como em:

```
joao = DOMINIO\gdh  
joao = DOMINIO\maria  
joao = DOMINIO\jose  
joao = DOMINIO\isac
```

Basta criar um arquivo de texto usando qualquer editor e salva-lo, prestando atenção no uso de barras invertidas.

Quando qualquer usuário especificado no arquivo é autorizado pelo PDC, o servidor Samba local realiza a leitura no sistema de arquivos utilizando a conta "joao", que é a única cadastrada localmente. Com isso, você precisa apenas manter o arquivo atualizado, sem se preocupar com senhas. Ao administrar uma rede com várias estações, é interessante manter o SSH ativo em todas as máquinas, de forma que você possa atualizar o arquivo remotamente, quando necessário.

As serem acessados através do ambiente de redes, os compartilhamentos das máquinas Linux ficam disponíveis para as demais máquinas do domínio sem necessidade de autenticação adicional, já que a autenticação é centralizada no PDC:



Concluindo, caso você deseje mais tarde remover a máquina Linux do domínio, basta alterar novamente o smb.conf (na estação), mudando a linha "workgroup = ", para que ela passe a indicar o nome do grupo de trabalho e não mais do domínio e alterar a linha "security = domain" para "security = user", como em:

```
[global]
workgroup = grupo
netbios name = M5
security = user
encrypt passwords = yes
```

Depois de reiniciar o Samba (ou aguardar o tempo de atualização após a mudança no arquivo), a estação deixa o domínio e volta a fazer parte do grupo de trabalho.

A principal limitação dessa configuração é que ela permite centralizar apenas a autenticação dos compartilhamentos de rede, mas resolve o problema da autenticação local nas estações Linux, que continua sendo feita da forma tradicional.

Usando o PDC para autenticação local

É possível configurar os clientes Linux para fazerem a autenticação dos usuários locais no PDC e armazenarem as configurações no próprio servidor, assim como no caso das máquinas Windows, mas nesse caso a configuração é bem mais complicada, pois temos

que fazer várias alterações que alteram a forma como sistema autentica os usuários. Ao invés de verificar os arquivos `"/etc/passwd"` e `"/etc/shadow"`, onde ficam armazenadas as contas locais, o cliente passa a utilizar o Samba e o Winbind para buscar os logins no servidor e assim autenticar o usuário.

Se você procura uma solução simples e limpa, recomendo que se limite à configuração que mostrei até aqui. Se não se importa de sujar as mãos, continue por sua conta e risco. :)

Esta configuração é indicada para distribuições derivadas do Debian que utilizam o KDM, ideal para situações em que você usa o Kurumin nos desktops da empresa e quer usar a lista de logins de um servidor Samba, ao invés de logins locais. Ela funciona em outras distribuições, mas eventualmente podem ser necessárias pequenas mudanças, de acordo com as peculiaridades de cada uma.

O primeiro passo é instalar os pacotes **"samba"** (ou `samba-server`), **"winbind"** (ou `samba-winbind`) e **"libpam-modules"** em cada cliente. Nas distribuições derivadas do Debian, instale diretamente os três pacotes:

```
# apt-get install samba winbind libpam-modules
```

No Fedora, o winbind está incluído no pacote principal do Samba e os módulos do PAM são instalados através do pacote `"pam_smb"`:

```
# yum install samba pam_smb
```

A configuração no servidor não muda em relação ao que já vimos. Toda a configuração que vemos aqui é feita nos clientes. Abra agora o arquivo `"/etc/samba/smb.conf"` (no cliente Linux) e faça com que a seção Global fique como o exemplo. Você pode tanto adicionar compartilhamentos, quanto ficar apenas com esta configuração básica:

```
[global]
netbios name = cliente1
workgroup = Dominio
winbind use default domain = yes
obey pam restrictions = yes
security = domain
encrypt passwords = true
wins server = 192.168.1.254
winbind uid = 10000-20000
winbind gid = 10000-20000
template shell = /bin/bash
template homedir = /home/%U
winbind separator = +
invalid users = root
```

Não se esqueça de substituir o "Dominio" pelo nome do domínio usado na rede, o "cliente1" pelo nome do cliente e o "192.168.1.254" pelo endereço IP do servidor Samba PDC.

Abra agora o arquivo `"/etc/nsswitch.conf"` e substitua as linhas:

```
passwd: compat
group: compat
shadow: compat
```

... no início do arquivo, por:

```
passwd: compat winbind
group: compat winbind
shadow: compat winbind
```

Um exemplo do arquivo completo é:

```
passwd: compat winbind
group: compat winbind
shadow: compat winbind
```

```
hosts: files dns mdns
networks: files
```

```
protocols: db files
services: db files
ethers: db files
rpc: db files
netgroup: nis
```

Depois de modificar os dois arquivos, reinicie o Samba e o Winbind e teste a configuração, ingressando no domínio. Para isso, use o comando "net rpc join":

```
# net rpc join member -U root
Password:
Joined domain DOMINIO.
```

A senha solicitada é a senha de root do servidor PDC, cadastrada no Samba, assim como fazemos ao cadastrar as máquinas Windows. Em caso de problemas, você pode usar também o comando abaixo, que especifica o nome do servidor (-S) e o nome do domínio (-w):

```
# net rpc join -S gdh -w dominio -U root
```

Se você receber uma mensagem de erro, como:

```
Creation of workstation account failed
Unable to join domain DOMINIO.
```

... provavelmente você esqueceu de cadastrar a máquina cliente no servidor. O nome da máquina (que você verifica através do comando "hostname") deve ser o mesmo que o incluído no arquivo smb.conf. Para criar a conta de máquina para o cliente, use (no servidor) os comandos que vimos anteriormente:

```
# useradd -d /dev/null -s /bin/false cliente1$
# passwd -l cliente1$
# smbpasswd -a -m cliente1
```

Neste ponto o cliente já está logado no domínio. Esta configuração é permanente, de forma que você não precisa se preocupar em refazer a configuração a cada boot.

Falta agora a parte mais problemática, que é configurar o PAM, o sistema de autenticação do sistema, para buscar os logins no servidor. Isso é feito modificando os arquivos `"/etc/pam.d/login"` e `"/etc/pam.d/kdm"`.

Comece adicionando as linhas abaixo no **início** do arquivo `"/etc/pam.d/login"` (responsável pela autenticação dos usuários no sistema), sem apagar as demais:

```
session required pam_mkhomedir.so skel=/etc/skel umask=0022
session optional pam_mount.so
auth sufficient pam_winbind.so
account sufficient pam_winbind.so
session required pam_winbind.so
```

Abra agora o arquivo `"/etc/pam.d/kdm"`, deixando o arquivo com o seguinte conteúdo (apague ou comente as demais linhas). A mesma configuração pode ser usada no arquivo `"/etc/pam.d/gdm"`, usado por distribuições que trazem o Gnome por padrão:

```
auth required /lib/security/pam_securetty.so
auth required /lib/security/pam_nologin.so
auth sufficient /lib/security/pam_winbind.so
auth required /lib/security/pam_pwdb.so use_first_pass shadow nullok
account required /lib/security/pam_winbind.so
session required /lib/security/pam_mkhomedir.so skel=/etc/skel umask=0022
```

Esta configuração faz com que o KDM exiba a lista de usuários cadastrados no servidor e permita que você faça login diretamente no domínio, sem passar pela autenticação local. É importante também desativar o autologin do KDE (ainda no cliente), no Centro de Controle do KDE > Administração do Sistema > Gerenciador de login.



Se você apenas adicionar as linhas acima no `/etc/pam.d/kdm`, mas não apagar as linhas que já existem no arquivo (que permitem a autenticação local), a tela do KDM vai exibir a lista de logins do servidor, mas vai recusar o login, dizendo que a senha está incorreta. Este é um dos erros de configuração mais comuns :).

Se você deixar disponível a opção "Bloquear sessão" do KDE, vai precisar editar também o arquivo `/etc/pam.d/kscreensaver`, para que ele também use as contas do servidor. Caso contrário, o usuário vai acabar tendo que reiniciar o X, cada vez que clicar por engano no ícone.



Adicione as duas linhas abaixo no início do arquivo (`/etc/pam.d/kscreensaver`), sem apagar as demais:

```
auth sufficient pam_winbind.so
auth required pam_unix.so shadow nullok
```

Para que esta configuração funcione, é importante que os usuários sejam cadastrados no servidor como usuários reais, usando o comando `"adduser"`, e não o `"adduser --disabled-login --no-create-home"` ou similar. Basicamente, é preciso que o usuário possa se logar no servidor, caso contrário também não vai conseguir se logar nas estações.

No cliente, acesse a pasta `/etc/rc5.d` e verifique se os links responsáveis por inicializar os serviços **samba**, **winbind** e **kdm** foram criados corretamente. Eles precisam ser carregados nessa ordem. No caso de distribuições que inicializam o KDM primeiro (como no caso do Kurumin), renomeie o link, de forma que ele seja inicializado por último, como em:

```
# mv /etc/rc5.d/S02kdm /etc/rc5.d/S99kdm
```

Reinicie o cliente, para que os módulos do PAM sejam atualizados e os serviços inicializados na ordem correta. Você notará que a tela de login do KDM passará a exibir os usuários cadastrados no servidor, ao invés dos usuários locais, sintoma de que está tudo funcionando.



Configurando desta forma, os usuários locais que forem eventualmente criados no terminal chegam a aparecer na lista, mas não é possível fazer login neles através do KDM (essa é justamente a idéia). Apesar disso, você pode se logar nos terminais remotamente (usando o root e outros logins locais) via SSH, quando precisar alterar as configurações.

No arquivo `"/etc/pam.d/login"`, incluímos a linha `"session required pam_mkhome.so skel=/etc/skel umask=0022"`. Ela faz com que a pasta `"/etc/skel"` (da estação) seja usada como um template para a criação dos diretórios home dos usuários que só existem no servidor.

A pasta `"/home"` (na estação) armazena apenas os arquivos que forem alterados em relação à pasta `"/etc/skel"`, simplificando os backups. Você pode configurar o servidor Samba instalado em cada estação para compartilhar o diretório home, com permissões de acesso apenas para o administrador da rede, de forma que você possa acessar o home de cada estação a partir do servidor e fazer backup periodicamente.

O `"/etc/skel"` é justamente uma pasta modelo, cujo conteúdo é copiado para o diretório home, sempre que um novo usuário é criado. As configurações padrão mudam muito de distribuição para distribuição. Esta configuração privilegia o uso das configurações padrão de cada distribuição, permitindo que você use diversas distribuições diferentes nos clientes, independentemente de qual esteja usando no servidor. O Fedora continua com cara de Fedora, o Slackware de Slackware, e assim por diante.

Samba, parte 4: Compartilhando impressoras no Samba

O Samba oferece suporte aos mais diferentes sistemas de impressão, incluindo o BSD, SYSV, AIX, HPUX, QNX, PLP e LPRNG. Antigamente, criar um simples compartilhamento de impressora no Samba era uma tarefa espinhosa, já que você

precisava verificar qual era o sistema de impressão usado na instalação do sistema e especificar os comandos de impressão manualmente na configuração do Samba, adicionado opções como estas na seção [global], ou na seção referente a cada compartilhamento:

```
printing = bsd
print command = /usr/bin/lpr -P%p %s; /bin/rm %s
lpq command = /usr/bin/lpq -P%p
lprm command = /usr/bin/lprm -P%p %j
queue pause command = /usr/sbin/lpc stop %p
queue resume command = /usr/sbin/lpc start %p
```

Com a popularização do **Cups**, tudo se tornou muito mais simples, pois você precisa apenas adicionar as opções "printing = cups" e "load printers = yes" na seção [global] do smb.conf e nada mais:

```
printing = cups
load printers = yes
```

Na verdade, nas versões recentes do Samba estas linhas nem mesmo são obrigatórias, pois o Cups já é o sistema de impressão usado por padrão e as impressoras disponíveis são carregadas por padrão quando o Samba encontra uma configuração válida no arquivo.



De qualquer forma, se você está usando alguma distribuição antiga, pode checar se a versão do Samba instalada inclui suporte ao Cups usando o comando "smbd -b", como em:

```
# smbd -b | grep CUPS
```

Ele deve responder:

```
HAVE_CUPS
```

Continuando, o primeiro passo para compartilhar a impressora é instalá-la no servidor, o que pode ser feito da forma tradicional, utilizando utilitários como o **kaddprinterwizard** (usado nas distribuições com o KDE) o **gnome-cups-add** (o utilitário equivalente no

Gnome) ou o **system-config-printer** (usado no Fedora) o que, desde que a impressora seja bem suportada, é bastante simples nas distribuições atuais:



Estas ferramentas de configuração estão fortemente atreladas às bibliotecas do KDE e do Gnome, de forma que elas não estarão disponíveis se você fizer uma instalação enxuta do sistema no servidor, sem instalar os ambientes gráficos.

Naturalmente, os desenvolvedores do Cups pensaram nessa possibilidade e adicionaram uma interface de administração via web, similar ao Swat, que pode ser usada até mesmo no caso de servidores sem interface gráfica, que você acessa remotamente:



A interface de administração fica acessível através da porta 631 (TCP) do servidor e pode ser acessada através do navegador, tanto localmente (através do endereço <http://127.0.0.1:631>) quanto remotamente (através do <http://servidor:631>). O grande problema é que você tem acesso às opções administrativas, como adicionar ou remover impressoras apenas ao acessar a interface localmente, o que é um problema quando você está configurando o servidor remotamente.

É possível alterar as permissões de acesso, de forma a liberar o acesso para o endereço IP do seu micro de forma simples editando o arquivo de configuração do CUPS, o `"/etc/cups/cupsd.conf"`. Procure a seção referente à pasta `"/admin"` (onde estão concentradas as opções administrativas) e adicione uma linha autorizando o endereço IP da sua máquina logo depois do `"Allow localhost"`, como em:

```
<Location /admin>
Order allow,deny
Allow localhost
Allow 192.168.1.10
</Location>
```

Depois da alteração, reinicie o serviço e você poderá acessar a interface sem limitações e assim fazer toda a configuração da impressora:

```
# /etc/init.d/cupsys restart
```

Depois de instalar e testar a impressora no servidor, o próximo passo é compartilhá-la através do Samba.

A forma mais simples de fazer isso é adicionar o compartilhamento "[printers]" no arquivo de configuração. Ele é um serviço interno do Samba, similar ao "[homes]", que permite compartilhar de uma vez todas as impressoras disponíveis no servidor e replica as mudanças na configuração do CUPS de forma automática.

O serviço "[printers]" pode ser inclusive usado em conjunto com o "[homes]", basta adicionar as duas seções no arquivo de configuração. A única observação ao usar os dois em conjunto é que você não pode ter um usuário e uma impressora com o mesmo nome, caso contrário o servidor não conseguirá compartilhar a impressora.

A principal vantagem de usar o "[printers]" é que você não precisa especificar manualmente quais impressoras deseja compartilhar, basta configurar as impressoras no CUPS e incluir a seção referente ao compartilhamento no smb.conf:

```
[printers]
comment = Todas as Impressoras
print ok = yes
guest ok = yes
path = /var/spool/samba
```

Um exemplo de arquivo completo, incluindo o compartilhamento, seria:

```
[global]
netbios name = Hades
workgroup = Grupo
server string = Servidor
encrypt passwords = true
preferred master = yes
os level = 100
preferred master = yes
wins support = yes
```

```
printing = cups
load printers = yes
```

```
[homes]
valid users = %S
create mask = 0700
directory mask = 0700
browseable = no
```

```
[arquivos]
path = /mnt/hda6
writable = no
write list = +arquivos
```

```
[printers]
comment = Todas as Impressoras
path = /var/spool/samba
print ok = yes
```

```
guest ok = yes  
browseable = yes
```

A opção "print ok" é similar à opção "available" que usamos nos compartilhamentos de pastas. Ao usar o "print ok = yes" a impressora fica disponível e, ao usar "print ok = no" o compartilhamento é desativado temporariamente. É obrigatório incluir esta opção no compartilhamento, pois é justamente ela que indica que trata-se de um compartilhamento de impressora.

A opção "guest ok = yes" indica que a impressora deve ficar disponível para o uso de qualquer um. Se preferir que ela fique disponível apenas para os usuários cadastrados no Samba, mude para "guest ok = no".

A opção "path" indica o diretório do sistema onde serão armazenados os trabalhos de impressão. A pasta "/var/spool/samba" é usada por padrão e deve ter sido criada automaticamente durante a instalação do Samba. De qualquer forma, se mais para a frente você não conseguir imprimir, recebendo mensagens de "disco cheio" ou "acesso negado" a partir dos clientes, verifique se a pasta realmente existe e se as permissões estão corretas:

```
# ls -l /var/spool/ | grep samba
```

Ele deve responder algo como:

```
drwxrwxrwt 2 root root 4096 2008-01-24 15:37 samba
```

O drwxrwxrwt indica as permissões da pasta, no caso uma pasta pública onde todos os usuários podem ler e gravar arquivos. O último "t" indica o uso do sticky bit, uma precaução de segurança, que faz com que cada usuário possa alterar apenas seus próprios arquivos. Isso evita que algum engraçadinho consiga corromper trabalhos de impressão enviados por outros usuários.

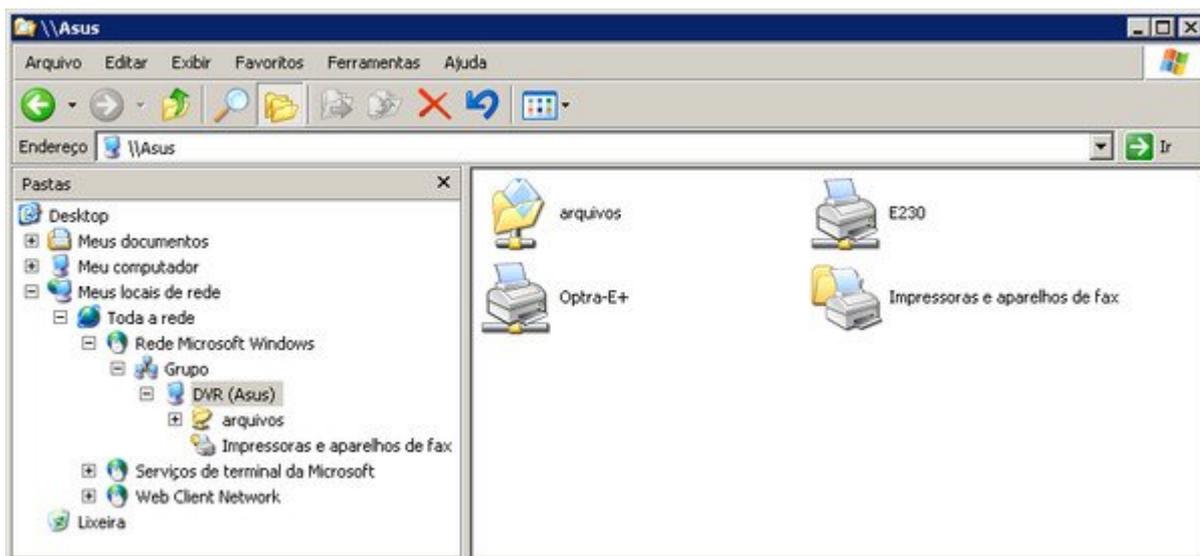
Se você precisar criar manualmente a pasta, o comando para setar as permissões corretamente é:

```
# chmod 1777 /var/spool/samba/
```

(note o uso do "1", que ativa o stick bit)

Continuando, depois de reiniciar o Samba, ou aguardar o tempo de atualização as impressoras passarão a aparecer no ambiente de redes, com os mesmos nomes que foram definidos ao instalar as impressoras no servidor.

O Samba pode inclusive ser usado para centralizar as impressoras da rede, recompartilhando impressoras disponibilizadas por outros micros, desde que você as configure corretamente no Cups. Nesse screenshot, por exemplo, temos duas impressoras. A "E230" está instalada diretamente no servidor, enquanto a "Optra-E+" é uma impressora disponibilizada por outro micro. Como pode ver, o cliente pode visualizar e imprimir em ambas:



É possível também especificar individualmente o compartilhamento de cada impressora, o que é útil quando o servidor compartilha várias impressoras diferentes e você precisa especificar as permissões individualmente. A configuração a adicionar no arquivo de configuração é praticamente a mesma. A principal diferença é que agora você deve especificar o nome da impressora no nome do compartilhamento, ao invés de usar a string "printers", como em:

```
[E230]
print ok = yes
guest ok = yes
path = /var/spool/samba
```

Assim como no caso dos compartilhamentos de arquivos, você pode limitar o acesso à impressora com base nos endereços IP ou nomes das máquinas, com base nos logins de usuário, ou através de uma combinação de ambos, através das opções "hosts allow", "hosts deny", "valid users" e "invalid users". Estas opções podem ser usados tanto ao ativar o serviço [printers] quanto ao compartilhar as impressoras individualmente.

Para permitir que que a impressora seja usada por apenas alguns endereços específicos você usaria:

```
[E230]
print ok = yes
guest ok = yes
path = /var/spool/samba
hosts allow = 192.168.1.3, 192.168.1.4, 192.168.1.65
```

Você pode também usar os nomes das máquinas dentro da rede windows no lugar dos endereços IP, como em:

```
[E230]
print ok = yes
guest ok = yes
```

```
path = /var/spool/samba
hosts allow = micro1, micro2, micro3
```

Para bloquear o acesso à impressora para os usuários "joao" e "maria":

```
[E230]
print ok = yes
guest ok = no
path = /var/spool/samba
invalid users = joao, maria
```

Para inverter a lógica, permitindo que apenas os dois usem a impressora:

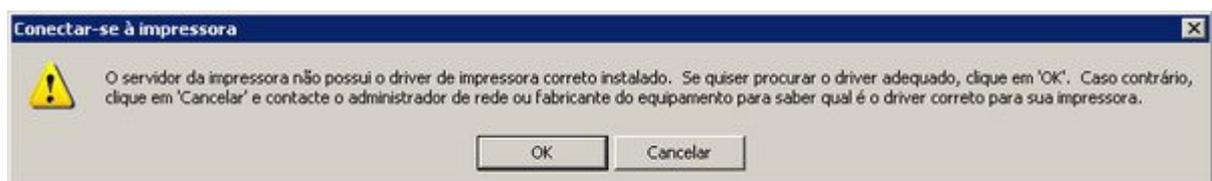
```
[E230]
print ok = yes
guest ok = no
path = /var/spool/samba
valid users = joao, maria
```

Para combinar as duas coisas, permitindo que a impressora seja usada apenas pelos dos usuários e, além disso apenas a partir de dois endereços específicos:

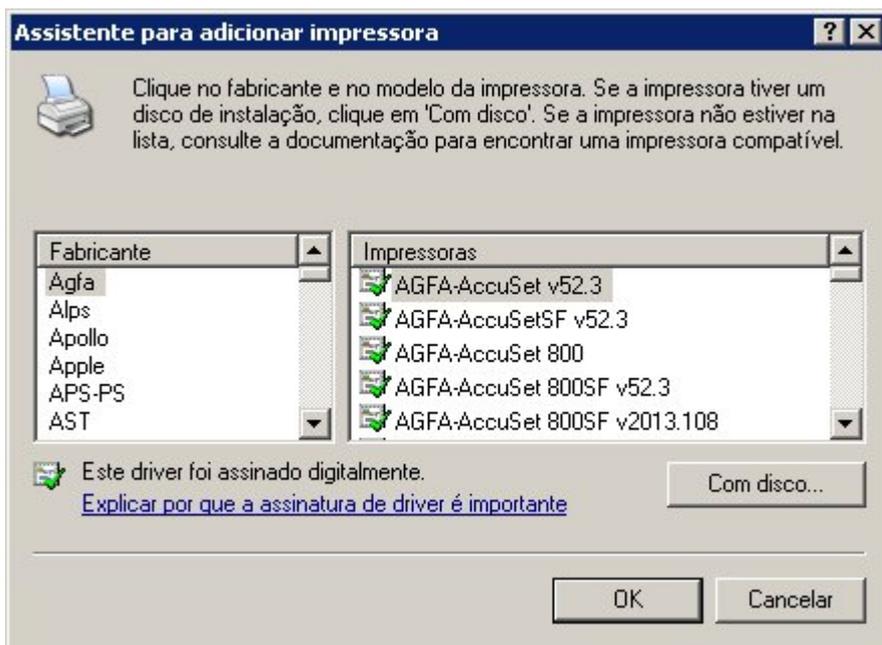
```
[E230]
print ok = yes
guest ok = no
path = /var/spool/samba
valid users = joao, maria
hosts allow = 192.168.1.3, 192.168.1.4
```

Continuando, a impressora pode ser instalada nos clientes Windows através do "Painel de Controle > Impressora > Adicionar Impressora > Impressora de rede" ou simplesmente clicando sobre ela no ambiente de rede. O Samba não se preocupa com o driver de impressão, apenas disponibiliza um spool remoto no qual os clientes podem colocar os trabalhos de impressão. Devido a isso, é necessário instalar os drivers de impressão nos clientes, da mesma forma que você faria ao instalar uma impressora local.

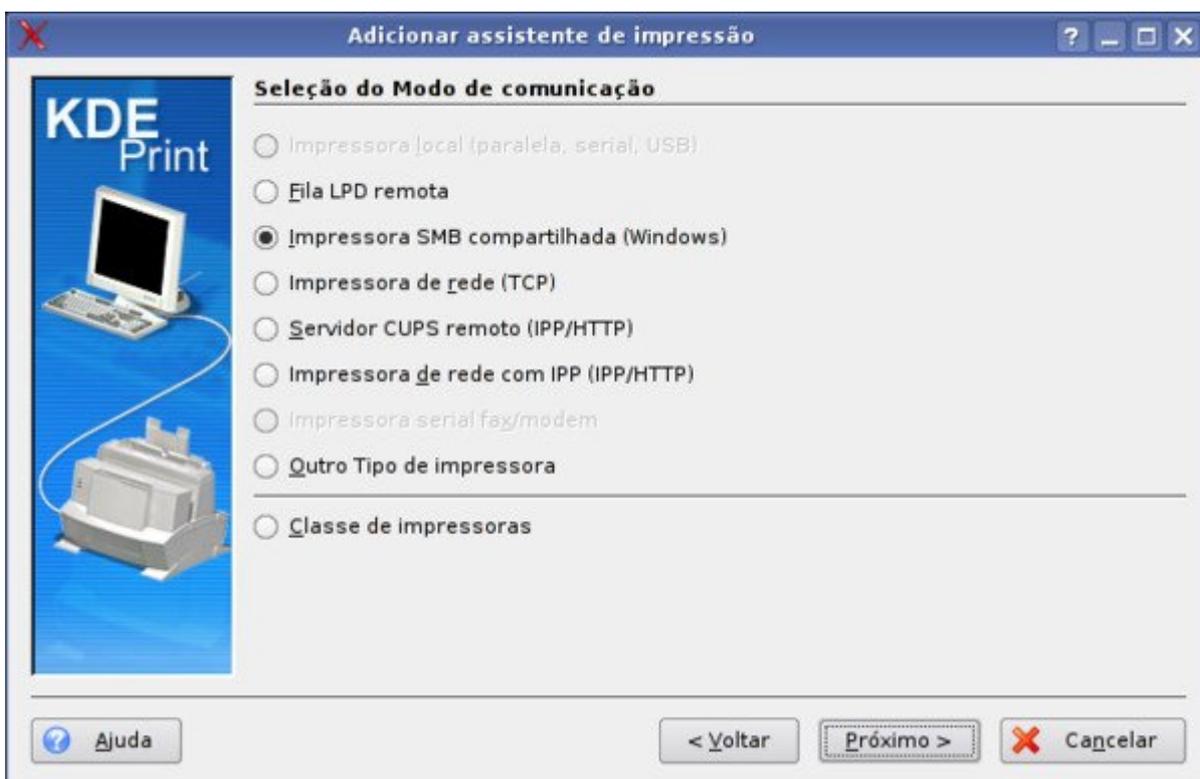
Inicialmente, você receberá uma mensagem de erro ao instalar a impressora nos clientes, avisando que o servidor não possui o driver instalado:

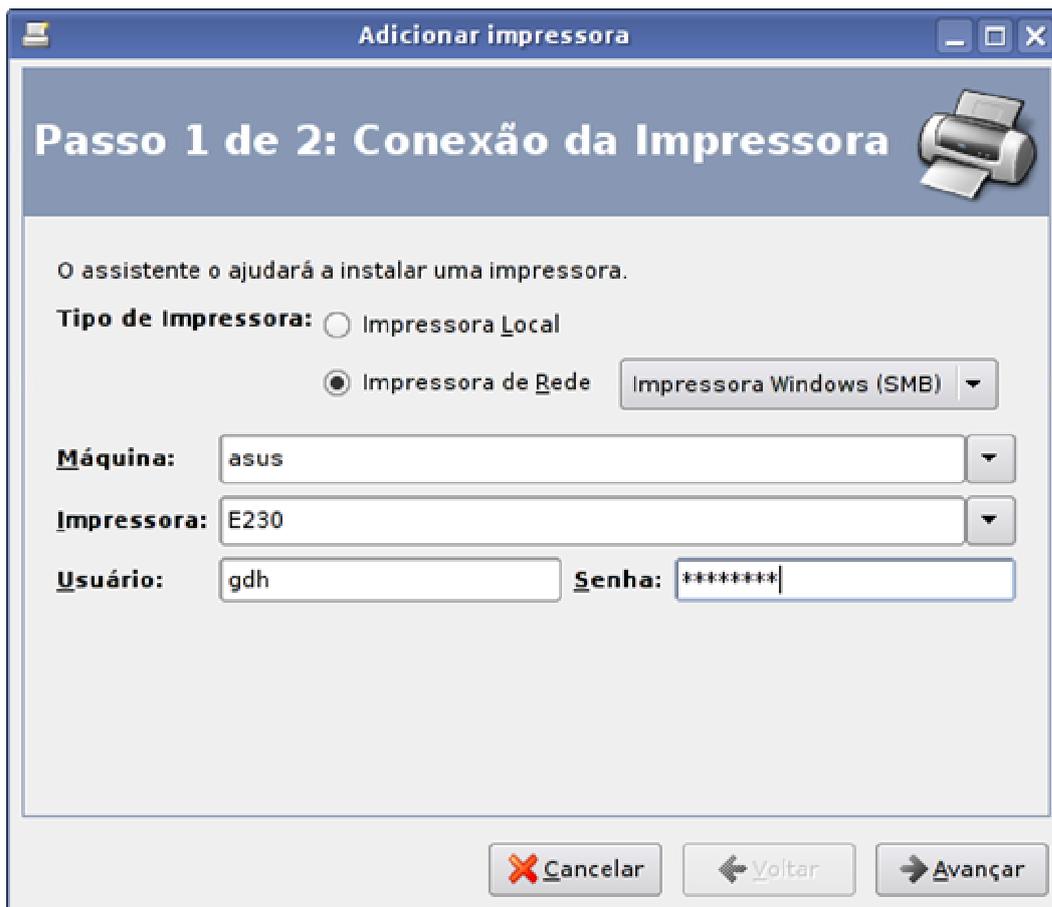


Esta mensagem se refere a outro recurso suportado por servidores Windows, onde você pode fazer o upload dos drivers de impressão para o servidor, de forma que os clientes possam obtê-los automaticamente ao se conectarem à impressora. Por enquanto ainda não configuramos isso, de forma que é preciso instalar a impressora da forma tradicional, fornecendo os drivers manualmente no cliente:



Naturalmente, as impressoras compartilhadas através do Samba podem também ser usadas a partir dos clientes Linux, que precisam apenas ter instalado o CUPS e o cliente Samba. Ao instalar a impressora nos clientes, procure pela opção de instalar uma impressora Windows ou SMB, que é suportada pela maioria das ferramentas de configuração. No caso do kaddprinterwizard você usaria a opção "Impressora SMB compartilhada (Windows)" e no gnome-cups-add a opção "Impressora Windows (SMB)":





É possível também instalar as impressoras nos clientes Linux diretamente via linha de comando usando o comando "lpadmin", como em:

```
# lpadmin -p E230 -E -v smb://192.168.1.254/E230
```

O parâmetro "-p" especifica o nome da impressora, conforme será instalada no cliente (não precisa necessariamente ser o mesmo nome usado pelo servidor). O "-v" indica a localização da impressora (endereço IP ou nome do servidor e o nome do compartilhamento), nesse caso estamos instalando a impressora "E230" compartilhada pelo servidor disponível no endereço 192.168.1.254.

Se o compartilhamento no servidor incluir a opção "guest ok = yes" você conseguirá acessar a impressora diretamente, caso contrário você precisará especificar o login e senha ao instalá-la. Nesse caso, o comando ficaria:

```
# lpadmin -p E230 -E -v smb://gdh:12345@192.168.1.254/E230
```

Veja que o login e senha são especificados diretamente no comando, entre o "smb://" e o endereço do servidor, que é agora separado por um "@".

Disponibilizando drivers de impressão para os clientes

Em uma pequena rede, instalar os drivers manualmente ao configurar a impressora nos clientes não seria um grande problema, já que você poderia simplesmente carregar o CD

de instalação, ou mesmo criar um compartilhamento de rede contendo os arquivos e fazer a instalação manualmente em cada um. Entretanto, em uma rede isso pode ser bastante tedioso.

Chegamos então recurso de upload de drivers de impressão que, naturalmente, também é suportado pelo Samba. Ele consiste em um compartilhamento oculto, chamado "print\$", que contém os drivers que serão fornecidos aos clientes.

Depois de configurar o recurso, o uso das impressoras nos clientes torna-se muito mais simples, pois você precisa apenas clicar sobre o ícone da impressora no "Meus locais de rede" para instalá-la, recurso chamado de "point and print" ou "p-n-p" (diferente do PnP, de "plug-and-play"). O Windows exibe um aviso, confirmando a instalação do driver e em seguida a impressora é instalada automaticamente:



Configurar este recurso é um pouco trabalhoso, mas não chega a ser difícil, vamos lá :).

O primeiro passo é criar um usuário administrativo, que você usará para acessar o servidor a partir dos clientes Windows e assim poder dar o upload dos drivers. Comece criando o usuário no servidor e cadastrando-o no Samba da forma tradicional:

```
# adduser gdh  
# smbpasswd -a gdh
```

O próximo passo é ativar o uso de privilégios (que vamos usar mais adiante) no Samba e criar um compartilhamento chamado "print\$", o compartilhamento oculto onde irão os drivers de impressão. Para isso, precisaremos fazer duas alterações no arquivo "/etc/samba/smb.conf".

A primeira é adicionar a linha "enable privileges = yes" no final da seção "[global]", sem alterar as demais, como em:

```
[global]  
workgroup = GRUPO  
netbios name = Asus  
server string = Servidor  
encrypt passwords = true  
wins support = yes  
preferred master = yes  
# invalid users = root  
os level = 100  
enable privileges = yes
```

Se você usou o swat para configurar o arquivo, muito provavelmente ele conterá a linha "invalid users = root". É importante que esta linha seja removida ou comentada (como no meu exemplo), caso contrário você não conseguirá atribuir os privilégios para o usuário, como faremos em seguida.

O próximo passo é incluir as linhas referentes ao compartilhamento "[printer\$]", que é um pouco diferente de um compartilhamento normal:

```
[print$]
comment = Drivers de impressão para os clientes Windows
path = /var/smb/printers
read only = yes
write list = gdh
inherit permissions = yes
```

A opção "path" diz qual a pasta do servidor onde serão colocados os drivers. Aqui estou usando a pasta "/var/smb/printers", mas você pode usar outra pasta se quiser.

Em seguida usamos a opção "read only = yes" para que o compartilhamento seja somente-leitura e usamos a opção "write list" para criar uma exceção, permitindo que o usuário administrativo que criamos na etapa anterior possa gravar no compartilhamento. A segurança é importante, pois os drivers são baixados automaticamente para os clientes Windows, de forma que alguém mal intencionado que pudesse alterar o conteúdo da pasta poderia muito bem usar o serviço como um vetor para transmitir vírus e spywares para os clientes Windows da rede.

Você pode também usar um grupo, como em "write list = @ntadmin" ou uma lista de usuários, como em "write list = gdh, admin", o importante é limitar o acesso apenas às pessoas autorizadas. Não se esqueça de reiniciar o Samba ou aguardar alguns minutos para que as alterações entrem em vigor.

O próximo passo é criar a pasta, criar as subpastas WIN40 (drivers para estações 95/98/ME) e W32X86 (estações com o NT/2000/XP) e ajustar as permissões, de forma que o usuário criado tenha permissão para alterar o conteúdo da pasta e os demais possam apenas ler:

```
# mkdir -p /var/smb/printers
# cd /var/smb/printers
# mkdir WIN40 W32X86
# chown gdh WIN40 W32X86
# chmod 2775 WIN40 W32X86
```

Falta agora uma etapa importante, que é transformar o usuário em um administrador de impressão no Samba, sem isso, ele terá acesso ao compartilhamento, mas não conseguirá dar upload dos drivers a partir dos clientes, usando o procedimento que veremos a seguir.

Isso é feito usando o comando "net", usado para ajustar os privilégios dos usuários do Samba, que deve ser executado no servidor, como root. Se o servidor se chama "asus" e o usuário se chama "gdh", o comando seria:

```
# net -S localhost -U root -W ASUS rpc rights grant 'ASUS\gdh'  
SePrintOperatorPrivilege
```

A opção "-S localhost -U root" diz que o comando net deve se conectar ao localhost, usando a conta de root. A opção "-W ASUS" especifica o nome do servidor (como definido na configuração do Samba) e o "grant 'ASUS\gdh' SePrintOperatorPrivilege" adiciona os privilégios para o usuário "gdh" do servidor "asus".

Ele vai pedir a senha de root e em seguida exibir uma mensagem de confirmação:

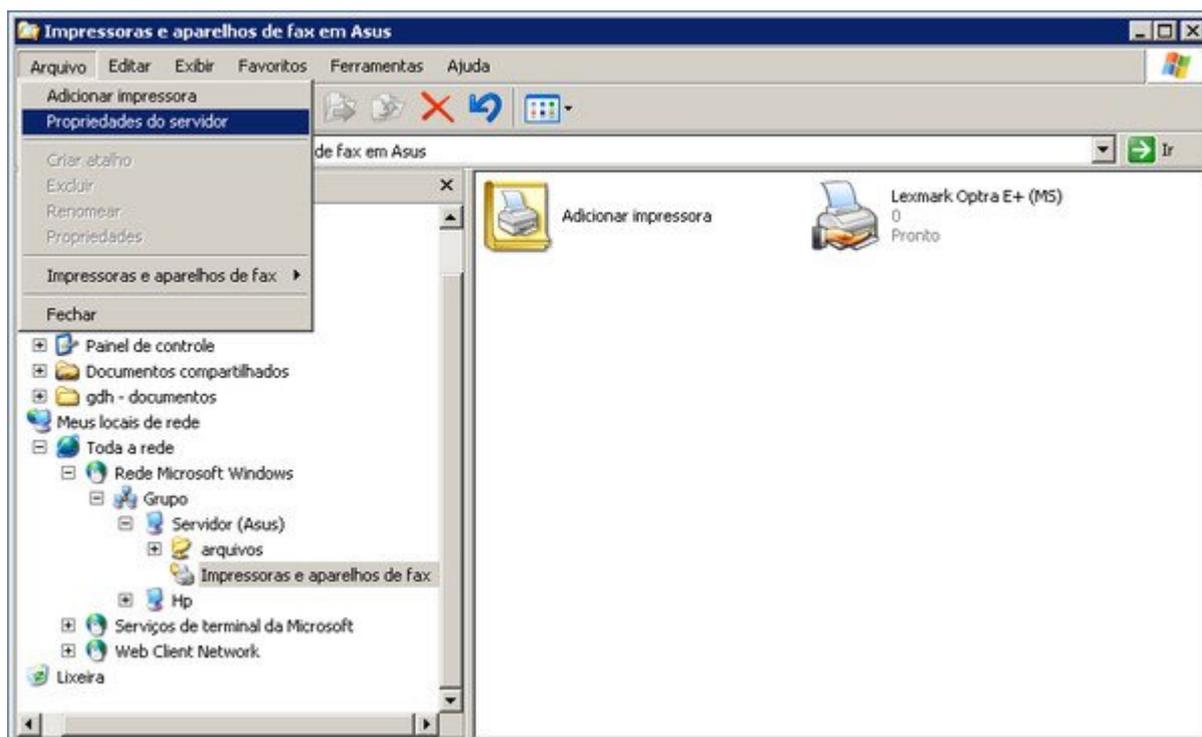
```
Password:  
Successfully granted rights.
```

Se nesse ponto você receber uma mensagem de erro, dizendo que não é possível se logar no servidor, muito provavelmente você se esqueceu de comentar a linha "invalid users = root", se esqueceu de adicionar a linha "enable privileges = yes" ou as alterações no arquivo ainda não entraram em vigor (nesse caso experimente reiniciar o Samba manualmente).

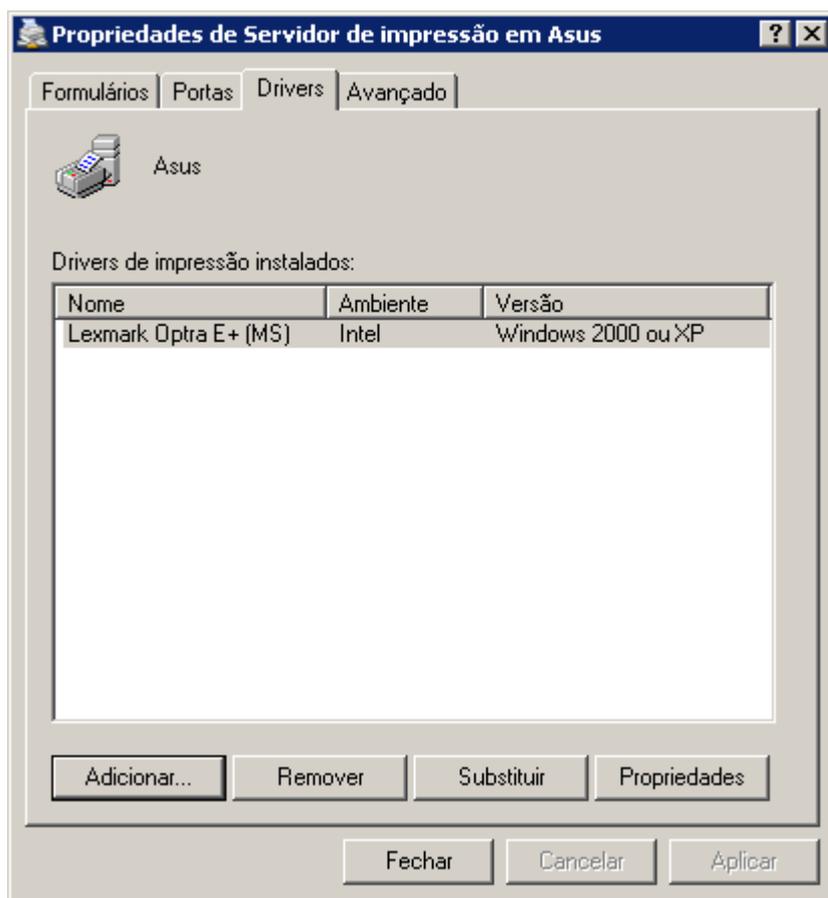
Com isso, concluímos a configuração no servidor. Os passos seguintes são feitos a partir de um cliente Windows da rede.

O primeiro passo é se logar no cliente usando o mesmo login (gdh no exemplo) que foi criado no servidor, já que apenas ele possui as permissões necessárias para atualizar os drivers. Caso necessário, adicione o usuário na estação usando o "Painel de Controle > Contas de usuário".

Acesse o servidor através do "Meus locais de rede", acesse a pasta "Impressoras e aparelhos de fax" e clique na opção "Arquivo > Propriedades do servidor" na janela principal do Explorer:



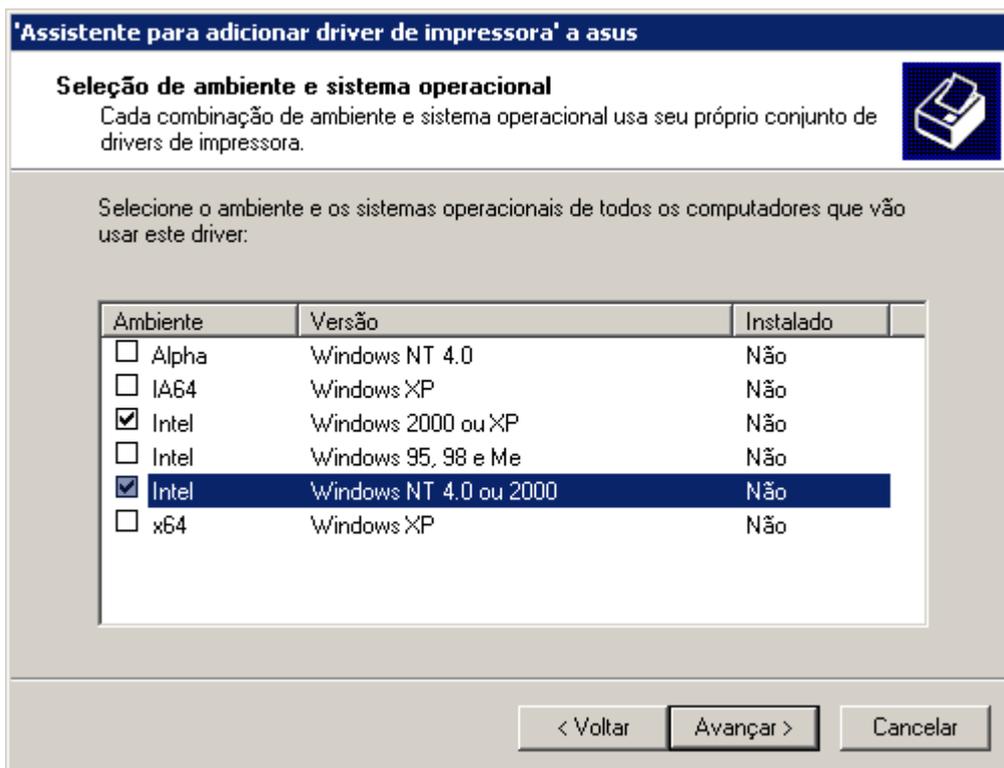
Na janela de propriedades, acesse a aba "drivers", que mostra os drivers disponíveis no servidor. Originalmente ela estará vazia. Use o botão "Adicionar" para instalar os drivers de impressão desejados:



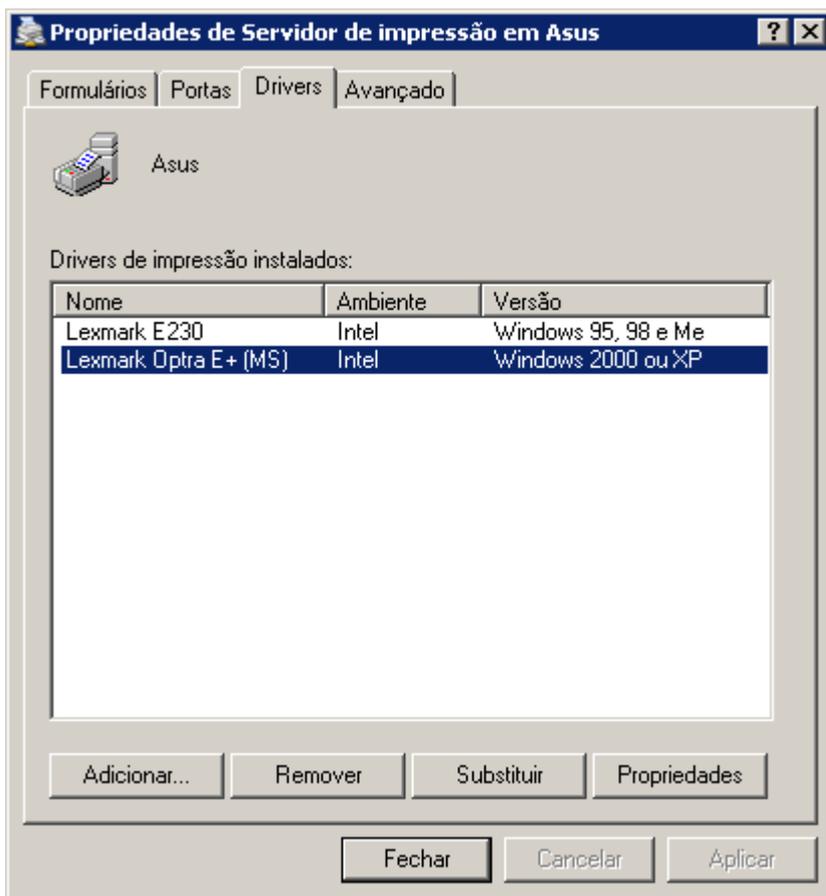
Se nesse ponto as opções não estiverem disponíveis, provavelmente você não adicionou o privilégio "SePrintOperatorPrivilege" para o usuário administrativo, ou não se logou usando o login correto na estação Windows.

Clicando no "adicionar" é aberta a tela padrão de seleção do driver, onde você pode usar um dos drivers do Windows ou especificar a localização de um driver. Entretanto, diferente do que teríamos normalmente, os drivers não são propriamente instalados, mas apenas copiados para o compartilhamento "print\$" do servidor.

A idéia da ferramenta é justamente permitir que você adicione vários drivers diferentes, que atendam clientes de diferentes versões do Windows, por isso, a cada driver, é aberta uma nova janela de seleção, que pergunta a que versões do Windows o driver é destinado:

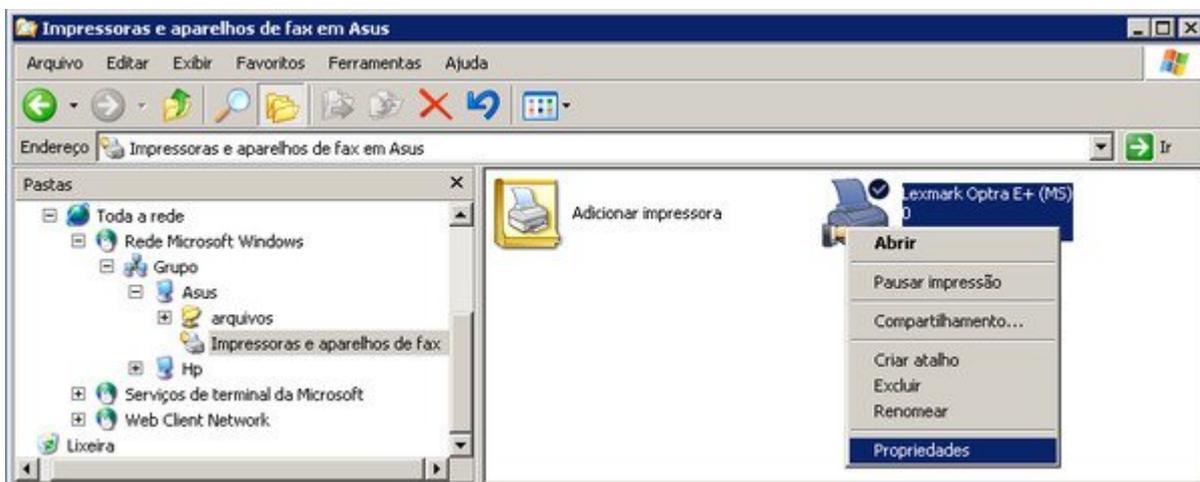


Dessa forma, você pode cadastrar um driver para máquinas com o Windows XP ou 2000, outra para os clientes com o 98/ME, outro para os com o XP de 64 bits e assim por diante. Se o servidor tiver mais de uma impressora instalada, você pode aproveitar para carregar os drivers das outras impressoras:

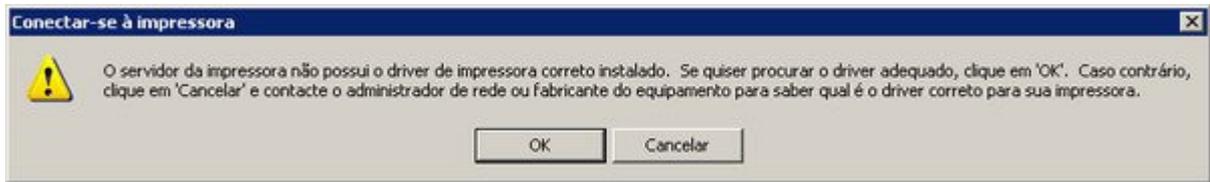


Nesse ponto, você verá que foram criadas subpastas dentro das pastas "/var/smb/printers/W32X86" e "/var/smb/printers/WIN40" do servidor, referentes aos drivers carregados.

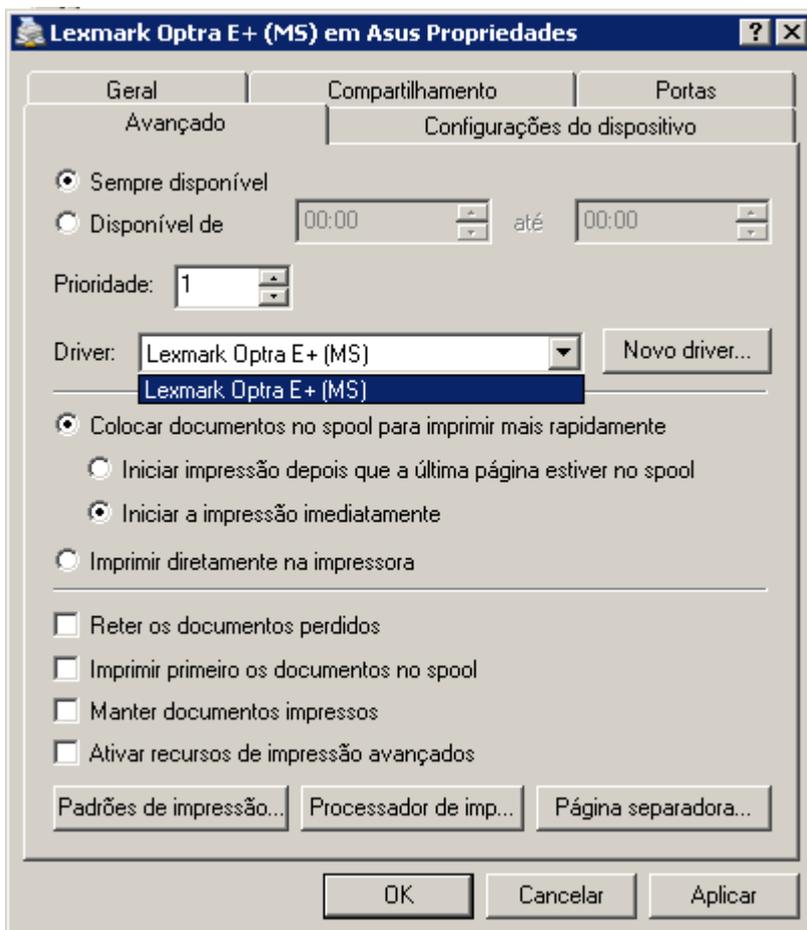
Por enquanto, os drivers foram apenas copiados para o servidor. É preciso ainda associar a impressora com o driver correspondente, para que o servidor passe a fornecê-lo para os clientes. Ainda logado com o usuário administrativo, clique com o botão direito sobre a impressora e acesse as propriedades:



Você receberá a mesma mensagem exibida ao instalar a impressora nos clientes, dizendo que o servidor não possui o driver de impressão (é justamente isso que estamos corrigindo, afinal :).



Nesse ponto, a resposta natural seria clicar no "OK", mas se você fizer isso você vai abrir a tela de seleção do driver e acabar fazendo uma instalação local dos drivers da impressora que não é o que queremos. Por estranho que possa parecer, a resposta correta aqui é o botão "**Cancelar**", o que o levará às propriedades da impressora:



Na aba "Avançado" especifique o driver que será usado na opção "Driver", que originalmente estará em branco. Com isso, o driver é associado com a impressora, fazendo com que o servidor passe a fornecê-lo para os clientes que se conectarem a ela, concluindo a configuração. Se o servidor tiver outras impressoras compartilhadas, faça o mesmo para as demais.

Estes passos parecem estranhos e pouco intuitivos, mas são os mesmos passos que você usaria para instalar os drivers em um servidor de impressão Windows. O Samba simplesmente implementa as mesmas funções.

Uma observação é que ativar o upload de drivers faz com que as impressoras compartilhadas, disponíveis na pasta "Impressoras e aparelhos de fax" sejam renomeadas para o nome "oficial" fornecido pelo driver. É por isso que a minha "E230" foi renomeada para "Lexmark Optra E+ (MS)". Se você não quiser que isso aconteça, adicione a opção "force printername = yes" na seção referente à impressora (ou na seção [printers]) do smb.conf, como em:

```
[E230]
print ok = yes
guest ok = yes
path = /var/spool/samba
force printername = yes
```

Depois que a alteração é aplicada, a impressora volta a ser compartilhada com o nome definido por você.

Vamos então a mais um exemplo de configuração, desta vez bem mais incrementado, incluindo o compartilhamento de impressoras, o compartilhamento para os drivers Windows, lixeira e outros recursos que vimos até aqui:

```
[global]
netbios name = Cartago
server string = Servidor
workgroup = Grupo
local master = yes
os level = 100
preferred master = yes
wins support = yes
map to guest = bad user
guest account = guest

vfs objects = recycle
recycle:keeptree = yes
recycle:versions = yes
recycle:repository = /mnt/sda2/trash/%U
recycle:exclude = *.tmp, *.log, *.obj, ~*.*, *.bak, *.iso
recycle:exclude_dir = tmp, cache

printing = cups
load printers = yes
enable privileges = yes

[lixeira]
path = /mnt/sda2/trash/%U
writable = yes
```

```
[printers]
path = /var/spool/samba
print ok = yes
guest ok = yes
browseable = yes
```

```
[print$]
path = /var/smb/printers
read only = yes
write list = gdh
inherit permissions = yes
```

```
[arquivos]
path = /mnt/hda2
writable = no
write list = +arquivos
```

```
[engenharia]
path = /mnt/sda1/engenharia
writable = yes
valid users = +engenheiros
browseable = no
```

```
[gerencia]
path = /mnt/sda1/gerencia
writable = yes
valid users = joao, maria
hosts allow = 192.168.1.2, 192.168.1.32
browseable = no
```

```
[publico]
path = /mnt/sda2/publico
writable = yes
guest ok = yes
```

```
[backup]
path = /mnt/sda2/backup/%U
writable = yes
valid users = %U
writable = yes
guest ok = no
```

Este exemplo de configuração exige alguns passos adicionais para ser usado, incluindo a configuração das impressoras e a instalação dos drivers de impressão a partir dos clientes Windows, como vimos até aqui.

O exemplo inclui também o uso da lixeira para todos os compartilhamentos e o uso de 5 compartilhamentos de arquivos. Um deles é o compartilhamento "arquivos", que já utilizei em exemplos anteriores. Os arquivos ficam disponíveis para todos os usuários, mas apenas os usuários cadastrados no grupo "arquivos" podem fazer alterações. Continuando, temos os compartilhamentos "engenharia" e "gerencia", que são um pouco

mais seguros, acessíveis apenas para alguns usuários. Eles são também protegidos pela opção "browseable = no", que, como vimos, faz com que eles não sejam listados no ambiente de redes. Os três são complementados pelo compartilhamento "publico", que fica acessível para todos os usuários através do uso da conta "guest".

Este exemplo não inclui o [homes], que substituí por um compartilhamento para armazenamento de backups. Ele utiliza a variável "%U" (nome do usuário) para criar pastas particulares, onde cada usuário pode armazenar seus backups. Para usá-lo, seria necessário criar diversas subpastas dentro da pasta "/mnt/sda2/backup", uma com o nome de cada usuário e ajustar as permissões para que o usuário tenha acesso apenas à sua própria pasta, como em:

```
# mkdir /mnt/sda2/backup/joao
# chown joao.joao /mnt/sda2/backup/joao
```

Todos os usuários verão o compartilhamento "backup" ao acessarem o servidor, mas devido ao uso da variável, cada um verá apenas sua própria pasta ao acessá-lo.

Compartilhando através do Cups

o compartilhar impressoras, o Samba atua mais como um spool de impressão do que como um servidor propriamente dito, já que o trabalho pesado é na verdade feito pelo servidor Cups rodando abaixo dele.

Se você está configurando um servidor Samba, é natural usá-lo para compartilhar também as impressoras, já que o Samba oferece diversas opções de controle de acesso e outras opções avançadas, mas o próprio Cups possui um recurso nativo de compartilhamento de impressoras, que além de atender outras máquinas Linux (como seria de se esperar) permite que as impressoras sejam usadas também pelos clientes Windows, de uma forma bastante simples.

Para habilitar o compartilhamento, edite o arquivo `"/etc/cups/cupsd.conf"`, deixando-o com o seguinte conteúdo:

```
Port 631
Listen 631
Browsing On
BrowseAllow All
BrowseInterval 30
BrowseAddress @LOCAL
BrowseInterval 30
```

```
<Location />
Order allow,deny
Allow all
</Location>
```

```
<Location /printers>
Order allow,deny
Allow all
</Location>
```

```
<Location /admin>
Encryption Required
Order allow,deny
Allow localhost
</Location>
```

```
<Location /admin/conf>
AuthType Basic
Require user @SYSTEM
Order allow,deny
Allow localhost
</Location>
```

Veja que a seção `"/printers"`, que contém as impressoras, fica com permissão de acesso para todo mundo, enquanto o utilitário de administração do Cups (seção `/admin`) continua acessível apenas localmente, através do endereço **`http://127.0.0.1:631`**.

No caso do **Ubuntu**, a configuração de portas vai num arquivo separado, o `"/etc/cups/cups.d/ports.conf"`. Edite-o, substituindo a linha:

```
Listen localhost:631
```

Por:

```
Listen 631
```

Até aqui, não estamos impondo nenhum tipo de restrição, por isso contamos com o firewall para bloquear qualquer tentativa de impressão proveniente de micros da Internet. Você pode também fazer o compartilhamento de uma forma mais segura, especificando manualmente a faixa de endereços da rede local, ou mesmo especificando individualmente os endereços IP que poderão imprimir. Neste caso, as seções `<Location />` (onde vai a configuração que permite aos clientes verem as impressoras disponíveis) e `<Location /printers>` ficaria:

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 192.168.0.*
</Location>
```

```
<Location /printers>
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 192.168.0.*
</Location>
```

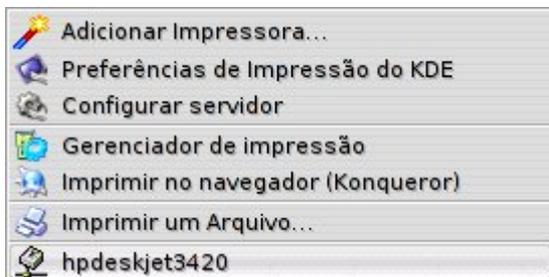
Não se esqueça de incluir o endereço `"127.0.0.1"` na lista. Caso contrário, todo mundo vai imprimir na impressora, menos você mesmo :).

Além da configuração simples, outra vantagem de compartilhar através do CUPS é que as impressoras podem ser configuradas automaticamente nos clientes Linux, sem necessidade de qualquer configuração manual. Basta manter a opção "browsing" ativa na configuração do CUPS nos clientes.

A opção browsing faz com que os clientes Linux da rede reconheçam automaticamente a impressora compartilhada e a configurem automaticamente durante o boot, sem necessidade de nenhuma intervenção manual. É um recurso bastante interessante: você dá boot usando uma distribuição Live-CD no cliente, manda imprimir qualquer coisa e o trabalho é direcionado de forma automática para a impressora compartilhada no servidor, sem que você precise fazer nada para configurá-la.

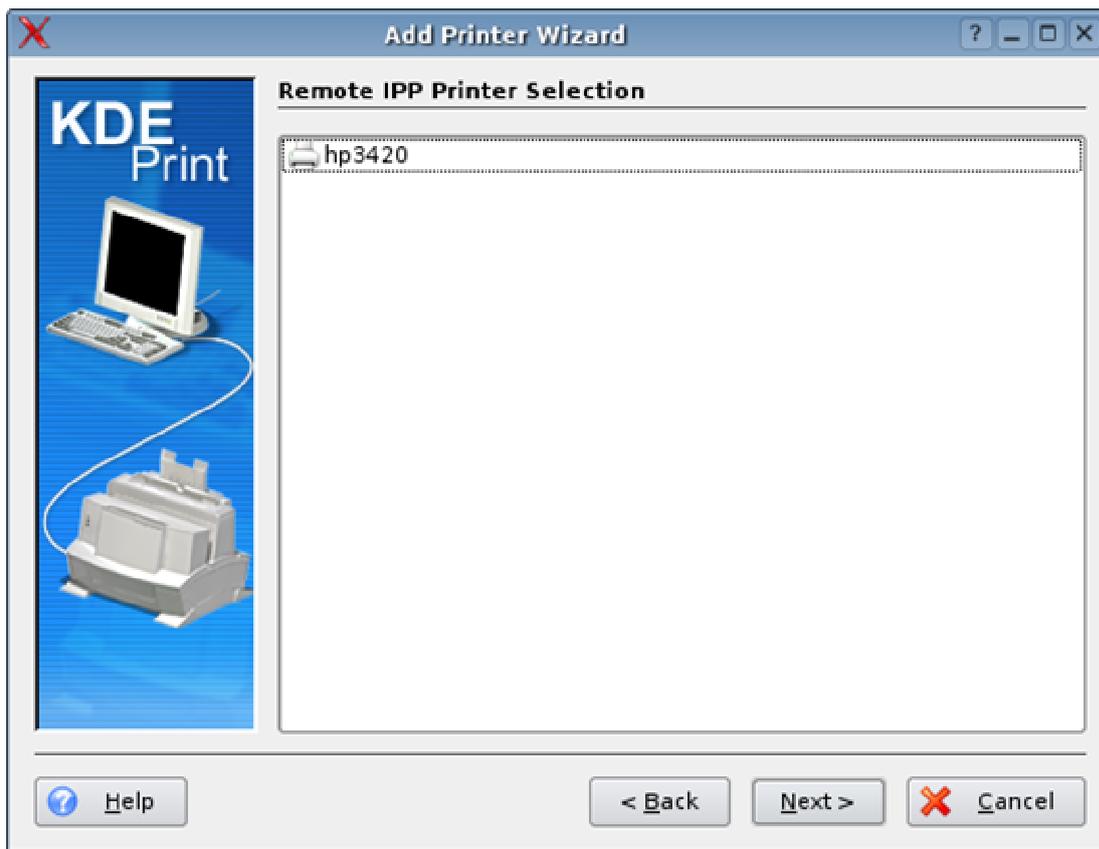
Funciona mais ou menos assim: durante o boot, o cliente manda um broadcast para a rede, perguntando se alguém está compartilhando impressoras. O servidor responde que está compartilhando a "hp" e aproveita para transmitir detalhes, como o modelo e driver usado pela impressora, configuração de impressão, etc. 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. O pacote de broadcast é reenviado periodicamente pelo cliente, permitindo que impressoras recentemente compartilhadas sejam descobertas.

Caso existam mais impressoras na rede, você pode escolher qual usar nas preferências de impressão. É um recurso que funciona surpreendentemente bem.



Caso você precise adicionar a impressora manualmente, abra o **kaddprinterwizard** e selecione a opção Remote CUPS Server. Forneça o endereço IP do servidor na rede local (ex: 192.168.0.10) e a porta onde o CUPS está escutando, que por padrão é a **631**.

Isso mostrará uma lista das impressoras disponíveis no servidor. Basta escolher a que será usada, apontar o driver que será usado e configurar as opções da impressora (papel, qualidade de impressão, etc.).



Nos clientes **Windows**, a configuração é semelhante. Eles não suportam o autodiscover, por isso é preciso adicionar a impressora manualmente pelo Painel de Controle > Impressoras e fornecer o CD com os drivers.

Vamos por passos. Comece abrindo o navegador e tentando acessar a página de administração do Cups no servidor. Acesse o <http://192.168.0.10:631> substituindo o "192.168.0.10" pelo endereço IP correto do servidor. 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>", na barra do navegador. Este é o endereço "completo" da sua impressora, que vamos usar na instalação.

De volta ao "Painel de Controle > Impressora", clique no "**Adicionar Impressora**" e marque a opção "**Impressora de rede**". Selecione a opção "Conectar-se a uma impressora na internet ou na intranet" e preencha o campo "URL" com o endereço completo da impressora (o "<http://192.168.0.10:631/printers/hp>" que anotamos no passo acima).

Se você estiver usando o Windows 2000 sem o Service Pack 2 ou o XP sem atualizações, ele vai dar um erro estúpido, dizendo que não é possível se conectar à impressora, mas isso é esperado. Dê ok e volte à tela inicial. Marque agora a opção "**Impressora local**" e deixe marcado o "Detectar e 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, etc.

Se você tem um servidor de impressão problemático na sua rede, que precisa ser reiniciado várias vezes ao dia, etc., recomendo que experimente substituí-lo por um servidor de impressão Linux. O Cups é um servidor de impressão muito sólido, ele raramente dá problemas. Uso na minha rede interna e até hoje não precisei reiniciar os micros por problemas na impressão uma única vez.

Se você estiver rodando o Windows em uma janela do VMware, o procedimento de instalação da impressora é o mesmo. Basta compartilhar a impressora no Linux e instalá-la no Windows do VMware seguindo os passos que mostrei acima, como se fosse uma impressora de rede.

Lembre-se de que qualquer tipo de compartilhamento de rede é sempre um risco potencial de segurança. Se você for ativá-lo em um micro simultaneamente conectado à internet e à rede local, não se esqueça de habilitar o firewall, abrindo apenas para os endereços da rede local.

O suporte a impressoras de rede compartilhadas no Cups foi incluído apenas a partir do Windows 2000. Para usar este recurso no Windows 95, 98 ou ME, você deve instalar o "Internet Printer Services", uma atualização disponibilizada pela Microsoft, que você pode baixar em:

<http://www.microsoft.com/windows98/downloads/contents/WUPreviews/IPP/Default.asp>

Depois de reiniciar, acesse o Painel de Controle > Impressora, clique no "Adicionar Impressora" e marque a opção "Impressora de rede". Coloque o endereço da impressora (<http://192.168.0.10:631/printers/hp>, por exemplo) no lugar do caminho para a impressora e forneça o driver.

Como criar um firewall e compartilhar conexão usando IPtables

Todo administrador de redes aprende logo que uma das coisas mais importantes para qualquer rede é um bom firewall. Embora existam muitos mitos em torno disto, os firewall não fazem milagres, apenas adicionam uma camada extra de proteção, escondendo as vulnerabilidades das máquinas. Você pode ter um servidor IIS ativo com todas as vulnerabilidades possíveis dentro da sua rede, mas ninguém poderá fazer nada se não conseguir se conectar a ele. Este é o papel do firewall, limitar e filtrar os acessos aos servidores e estações de trabalho da sua rede.

Existem vários tipos de firewall, de todos os preços. O tipo mais simples e ao mesmo tempo um dos mais eficazes para PCs domésticos são os firewalls de bloqueio, onde você simplesmente fecha todas as portas do micro (ou deixa abertas apenas as portas de que você realmente precisa). Se ninguém consegue se conectar a seu PC, 90% das brechas de segurança são anuladas.

Outro ponto comum é a necessidade de compartilhar a conexão com a Web. Nos meus artigos sobre o Coyote mostrei como usar um 486 para esta tarefa, desta vez vamos ver como é fácil fazer o mesmo com qualquer distribuição Linux. Isto permite que você use o seu próprio PC, sem precisar montar e manter outro micro só para isso, além de resolver as limitações do Coyote com modems PCI e placas de rede Wireless.

Isso pode ser feito facilmente através do Iptables. A receita funciona em qualquer distribuição que utilize o Kernel 2.4, basicamente qualquer coisa que você ainda possa querer usar hoje em dia.

Existem vários programas gráficos para configuração de firewalls, como por exemplo o GuardDog e o Shorewall (usando no Red Hat e Mandrake). Estes programas também trabalham com o Iptables, eles servem apenas para facilitar a configuração, criando as regras a partir das escolhas feitas pelo usuário.

A configuração do Iptables é feita diretamente via terminal, basta você ir inserindo as regras uma a uma. As regras se perdem ao reiniciar o micro por isso depois de testar tudo vamos criar um script para que elas sejam recriadas automaticamente a cada reboot.

O Iptables é tão versátil que pode ser usado para praticamente tudo relacionado à inspeção, encaminhamento e até mesmo alteração de pacotes. Se ele não fizer algo é possível criar um módulo que o faça. Já que as possibilidades são infinitas mais seu tempo não, vou ficar em algumas regras simples que resolvem a maior parte dos problemas do dia a dia. A partir daí você pode ir se aperfeiçoando e desenvolvendo soluções mais sofisticadas.

Antes de mais nada você precisa verificar se o pacote do iptables está instalado. Se você estiver no Mandrake basta dar um "**urpmi iptables**". Se você estiver no Debian, Kurumin ou Conectiva, um "**apt-get install iptables**" resolve.

Para garantir que o Iptables está mesmo carregado, dê também um:

```
modprobe iptables
```

Vamos então à criação das regras que determinam o que entra e o que não entra na máquina. Se o seu micro está ligado apenas à internet, sem uma rede local, então são necessárias apenas duas regras para resolver o problema. Abra um terminal, logue-se como root e digite o comando:

```
iptables -A INPUT -p tcp --syn -j DROP  
iptables -A INPUT -i ppp0 -p udp --dport 0:30000 -j DROP
```

Isso fará com que o micro passe a ignorar conexões vindas em qualquer porta TCP, sem enviar sequer uma confirmação de que o pacote foi recebido. Você continuará conseguindo acessar a internet normalmente, mas ninguém conseguirá se conectar diretamente ao seu PC; um servidor Web ou SSH que você esquecesse de desativar passariam despercebidos. Apenas as conexões iniciadas por você são aceitas, o que permite que alguns programas de compartilhamento como o gtkgnutella e o Kazza continuem funcionando normalmente. A segunda regra é opcional (dica do Fabricio Carvalho), ela bloqueia também parte das portas UDP, adicionando uma camada extra de segurança.

O efeito colateral é que alguns programas que abrem servidores podem deixar de funcionar. Você não conseguirá mais receber arquivos pelo ICQ por exemplo, como se estivesse acessando através de uma conexão compartilhada via NAT.

O interessante é que você pode desativar o firewall a qualquer momento, para isso basta um único comando:

```
iptables -F
```

Isso elimina todas as regras do Iptables, fazendo com que seu micro volte a aceitar todas as conexões. Você pode usa-la para permitir que alguém se conecte rapidamente via ssh na sua máquina por exemplo e depois fechar tudo novamente reinserindo as regras anteriores.

Se você tiver uma rede local e quiser que os micros da rede interna seja capazes de se conectar normalmente, mas mantendo o bloqueio a tudo que vem da internet, basta dar um "iptables -F" e começar de novo, desta vez adicionando primeiro a regra que permite os pacotes vindos da rede local:

```
iptables -A INPUT -p tcp --syn -s 192.168.0.0/255.255.255.0 -j ACCEPT
```

Em seguida vem os comandos anteriores:

```
iptables -A INPUT -p tcp --syn -j DROP
```

Altere o "192.168.0.0/255.255.255.0" para a faixa de endereços e máscara de sub-rede que estiver utilizando na sua rede. Este exemplo serve para redes que utilizam a faixa de 192.168.0.1 até 192.168.0.254.

O Iptables processa os comandos em seqüência. Então todos os pacotes passam pela primeira instrução antes de ir para a segunda. Quando um pacote vem de um dos endereços da rede local é imediatamente aceito, os demais vão para as duas últimas linhas e acabam recusados. É uma simples questão de sim ou não. A primeira linha diz sim para os pacotes da rede local enquanto as duas ultimas dizem não para todos os demais.

Imagine agora que você queira permitir ao mesmo tempo pacotes vindos da rede local e uma certa porta vinda da Internet, como por exemplo a porta 22 do SSH. Neste caso você adicionaria mais uma regra, mantendo as regras anteriores:

```
iptables -A INPUT -p tcp --destination-port 22 -j ACCEPT
```

```
iptables -A INPUT -p tcp --syn -s 192.168.0.0/255.255.255.0 -j ACCEPT
```

```
iptables -A INPUT -p tcp --syn -j DROP
```

```
iptables -A INPUT -p udp -j DROP
```

Agora tudo o que vem na porta 22 (tanto da Internet quanto da rede local) é aceito, tudo o que vem da rede local é aceito e todo o resto é rejeitado. Você pode adicionar mais linhas para abrir outras portas. Se você quisesse abrir também as portas 1021 e 1080, a lista ficaria assim:

```
iptables -A INPUT -p tcp --destination-port 22 -j ACCEPT
```

```
iptables -A INPUT -p tcp --destination-port 1021 -j ACCEPT
```

```
iptables -A INPUT -p tcp --destination-port 1080 -j ACCEPT
```

```
iptables -A INPUT -p tcp --syn -s 192.168.0.0/255.255.255.0 -j ACCEPT
```

```
iptables -A INPUT -p tcp --syn -j DROP
```

Isso permite que você mantenha disponíveis apenas os servidores que você realmente quer disponibilizar e nos momentos que quiser. A qualquer tempo você pode dar um `iptables -F` e readicionar apenas as regras para fechar tudo.

Vamos então à segunda receita, para compartilhar a conexão. Ela é ainda mais simples e também permite ativar ou desativar o compartilhamento a qualquer momento.

Em primeiro lugar você deve configurar as suas placas de rede e modem e verificar se tanto a conexão com a Internet quando a conexão com os micros da rede local estão funcionando normalmente. O compartilhamento da conexão em si pode ser feito com apenas três comandos:

Para compartilhar a conexão do modem com a rede local:

```
modprobe iptable_nat
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Para compartilhar uma conexão via ADSL ou cabo instalada na eth0:

```
modprobe iptable_nat
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Para desativar o compartilhamento, você deve usar o comando :

```
iptables -t nat -F.
```

Isso mesmo, é só isso... :-) O compartilhamento é ativado ou desativado imediatamente, sem que seja necessário reiniciar a conexão. Rápido, prático e confiável.

As três linhas respectivamente ativam o módulo nat do iptables, responsável pela tradução de endereços, avisam para o iptables que ele deve direcionar todas as conexões recebidas para a interface ppp0 (o modem) ou eth0 (a primeira placa de rede) e devolver as respostas para os clientes e confirmam a ativação no arquivo de configuração do TCP/IP.

Não faz mal se você acessa via modem e não fica permanentemente conectado. A regra mantém o compartilhamento ativo mesmo que você desconecte e reconecte várias vezes.

Se os clientes da rede já estiverem configurados para acessar a web através do endereço IP usado pelo servidor (192.168.0.1 se você quiser substituir uma máquina Windows compartilhando através do ICS) você já deve ser capaz de acessar a web automaticamente nos demais PCs da rede.

Uma observação é que estas regras não incluem um servidor DHCP, você deve configurar os clientes com endereço IP fixo ou então ativar o serviço **DHCPD** na sua distribuição. No Mandrake ou Red Hat basta ativar o serviço no painel de controle e o DHCP já irá funcionar automaticamente.

A configuração nos clientes fica: Endereço IP: Qualquer endereço dentro da faixa de endereços usada pelo servidor. Ex: 192.168.0.3 Servidor DNS: Os endereços dos servidores DNS do seu provedor. Ex: 200.177.250.10 Gateway Padrão: O endereço do servidor. Ex: 192.168.0.1 Domínio: O domínio do seu provedor. Ex: terra.com.br

As linhas de compartilhamento da conexão não conflitam com as regras de firewall que vimos anteriormente, você deve apenas ter o cuidado de colocá-las no início da seqüência. Neste caso nosso script completo ficaria assim:

```
# Carrega os módulos
modprobe iptables
modprobe iptable_nat
```

```
# Compartilha a conexão
modprobe iptable_nat
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
# Abre algumas portas (opcional)
```

```
iptables -A INPUT -p tcp --destination-port 22 -j ACCEPT
iptables -A INPUT -p tcp --destination-port 1021 -j ACCEPT
iptables -A INPUT -p tcp --destination-port 1080 -j ACCEPT
```

```
# Abre para a rede local
```

```
iptables -A INPUT -p tcp --syn -s 192.168.0.0/255.255.255.0 -j ACCEPT
```

```
# Fecha o resto
iptables -A INPUT -p tcp --syn -j DROP
```

Se você quiser que o PC também não responda a pings, adicione a linha:

```
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Mais uma linha interessante de se adicionar, que protege contra pacotes danificados (usados em ataques DoS por exemplo) é:

```
iptables -A FORWARD -m unclean -j DROP
(esta linha deve ser adicionada antes das demais)
```

Agora já temos 10 comandos, fora os para abrir portas específicas. Não seria muito prático ficar digitando tudo isso cada vez que precisar reiniciar o micro. Para automatizar isso, basta colar todos os comandos dentro de um arquivo de texto. Você pode salvá-lo como por exemplo: /usr/local/bin/meu_firewall

Em seguida, dê permissão de execução para o arquivo (chmod +x /usr/local/bin/meu_firewall) e você terá um shell script que pode ser chamado a qualquer momento. Basta digitar:

```
meu_firewall
```

Para tornar a inicialização realmente automática, você precisa apenas colocar o comando num dos arquivos de inicialização do sistema. Abra o arquivo `/etc/rc.d/rc.local` e adicione a linha:

```
/usr/local/bin/meu_firewall
```

No Debian e Kurumin você pode usar o arquivo `/etc/init.d/bootmisc.sh`

As regras que vimos acima funcionam como um firewall de bloqueio. Ou seja, o servidor não deixa que ninguém acesse os compartilhamentos de arquivos ou conectem o backoffice instalado na máquina com o Windows 98, mas não impedem que os usuários baixem e-mails com vírus ou que acessem uma página web que explore alguma das vulnerabilidades do IE por exemplo. Ao usar clientes Windows o ideal é complementar o firewall com um bom antivírus.

Vamos agora a um último passo que é forwardar certas portas para os hosts da rede interna. Isso permite que você rode um servidor de FTP ou crie um servidor de Counter Strike por exemplo em qualquer um dos micros da rede e não apenas no servidor que está diretamente conectado à internet. O servidor simplesmente direciona todas as requisições recebidas na porta para o micro especificado, de forma transparente. Também aprendemos a fazer isso no Coyote, lembra? Mas ele utiliza o ipchains, uma versão antiga do firewall, por isso os comandos são diferentes.

O forward de portas também usa o Nat, por isso você também deve carregar o módulo caso não tenha feito anteriormente:

```
modprobe iptable_nat
```

Em seguida vem as regras para fazer o forward da porta. Neste caso estou direcionando a porta 22 (do SSH) na conexão com a internet (eth0) para o micro 192.168.0.2 da rede local:

```
iptables -t nat -A PREROUTING -i etho -p tcp --dport 22 -j DNAT --to-dest 192.168.0.2  
iptables -A FORWARD -p tcp -i eth0 --dport 22 -d 192.168.0.2 -j ACCEPT
```

Basta alterar a regra, adicionando a porta e a máquina da rede interna para onde ele deve ser redirecionada. Se você acessa via modem, basta substituir o "eth0" em ambas as linhas por "ppp0". Esta regra pode ser usada em conjunto com as anteriores, mas deve ir sempre logo no início do arquivo, antes das regras para compartilhar a conexão e, claro, antes das regras para fechar tudo :-)

Você pode repetir o comando várias vezes para direcionar varias portas diferentes para várias máquinas. Naturalmente uma mesma porta não pode ser forwardada duas vezes.

Também é possível forwardar ranges de portas. No Unreal Tournament por exemplo você precisa abrir as portas UDP 7777, 7778 e 7779 neste caso as regras seriam:

```
iptables -t nat -A PREROUTING -i eth0 -p udp --dport 7777:7779 -j DNAT --to-dest 192.168.0.2
iptables -A FORWARD -p udp -i eth0 --dport 7777:7779 -d 192.168.0.2 -j ACCEPT
```

No bittorrent, que usa as portas tcp de 6881 a 6889 (ele tenta uma a uma até achar uma disponível) a regra seria:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6881:6889 -j DNAT --to-dest 192.168.0.2
iptables -A FORWARD -p tcp -i eth0 --dport 6881:6889 -d 192.168.0.2 -j ACCEPT
```

Neste link você encontra uma longa lista de portas usadas por vários aplicativos e jogos. Basta forwardá-las no servidor para que os clientes da rede interna possam utilizá-los normalmente. A limitação neste caso é que apenas um cliente pode usar cada porta de cada vez, mas em alguns casos o aplicativo é programado para escutar em várias portas simultaneamente (como no caso do bittorrent) e basta distribuir as portas usadas entre os clientes da rede.

http://www.practicallynetworked.com/sharing/app_port_list.htm

Evite abrir muitas portas no seu firewall, abra apenas as portas de que você realmente precisar e sempre termine o script com o `iptables -A INPUT -p tcp --syn -j DROP` para esconder todas as demais. Os famosos "buracos" no firewall surjem justamente de portas abertas que direcionam para programas ou máquinas vulneráveis. Você direciona a porta 1022 para um micro da rede interna com uma versão desatualizada do SSH, o invasor obtém acesso a ela e a partir aí tem uma base para lançar ataques contra outros micros da rede local, ataques muito mais efetivos diga-se de passagem, pois serão feitos de dentro, onde sua rede é vulnerável.